IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:		Atty. Docket No.:	004770.00026
	Toni Paila et al.		
Serial No.:	09/988,241	Group Art Unit:	2682
Filed:	November 19, 2001	Examiner:	West, Lewis G.
For:	MULTICAST SESSION HANDOVER	Confirmation No.:	8406

SECOND DECLARATION UNDER 37 C.F.R. § 1.131

The Honorable Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

We, Toni Paila (Finnish), Jani Poikela (Finnish), Lin Xu (Chinese), Juha-Pekka Luoma (Finnish), and Rod Walsh (British), hereby declare that:

1) We are the joint inventors of the above-captioned application, which is generally directed to Advanced Service Announcement for Broadcasting (ASAB);

2) Prior to August 20, 2001, the filing date of U.S. Patent No. 6,731,936 B2 (hereinafter "Chen"), we conceived of the invention recited in claims 1-47 of the above-captioned application, at least to the extent the claims are allegedly taught by Chen, and diligently pursued constructive reduction to practice in the form of a patent application filed with the United States Patent & Trademark Office.

3) Prior to August 20, 2001, we developed a protocol specification for Advanced Service Announcement for Broadcasting (ASAB), a version of which is attached as Exhibit A.

4) Correspondence at least as early as June 6, 2001 included a copy of the ASAB Protocol Specifications (see Exhibit B).

5) Concurrent to creating the ASAB Protocol Specifications, and also prior to August 20, 2001, we developed ASAB Server Side Specifications, a version of which is attached as Exhibit C.

6) Correspondence at least as early as July 30, 2001 included a copy of the ASAB Server Side Specifications (see Exhibit D).

7) Upon substantial completion of the specification documents attached as Exhibits A and C, we continued work on the development of ASAB, and prepared a disclosure document of an embodiment of the invention (Exhibit E). The disclosure document was submitted to the Nokia Internal Patent Committee at least as early as September 4, 2001, as evidenced in Exhibit E.

8) The Internal Patent Committee evaluates and processes received invention reports on a first come-first serve basis After receiving an invention report, the Internal Patent Committee performs a patent search for relevant prior art in order to facilitate the patent filing decision. If a decision is made to proceed with the preparation of a patent application based on the invention report, the invention is assigned a rating from 0 to 5 based on the potential value of a resulting patent, and an instruction letter is sent to an outside counsel, with the invention report, requesting preparation of a patent application for the invention.

9) After its in-turn review and analysis by the Internal Patents Committee, the disclosure document attached as Exhibit E was sent to our patent attorney, Mr. Bradley C. Wright of the law firm Banner & Witcoff, Ltd., on October 1, 2001, as evidenced by the email communication attached as Exhibit F.

10) On October 30, 2001, Ross Dannenberg (also an attorney with Banner & Witcoff, Ltd.) sent a draft of the above-captioned patent application to our employer for our review. A copy of the email communicating the draft is attached as Exhibit G.

11) On November 13, 2001 Ross Dannenberg sent a revised draft of the above-captioned patent application. A copy of the email communicating the revised draft is attached as Exhibit H.

12) On November 19, 2001, the above-captioned patent application was filed in the U.S. Patent and Trademark Office.

13) The exchange of draft applications with our patent attorney demonstrates diligence from before August 20, 2001 until the filing of the above-captioned patent application and the constructive reduction to practice of our invention.

14) All acts referred to in this Declaration were performed either in the United States, or in a WTO member country;

15) The attached Exhibits have not been altered since they were originally submitted to the Patent Committee or otherwise prepared or communicated; and

16) We declare under penalty of perjury under the law of the United States of America that statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Respectfully submitted,

Toni Paila	4/10/2004 Date
Jani Poikela	1/10/2004 Date
Lin Xu	21/09/04. Date
Luoma Juha-Pekka Luoma	30/09/04 Date
Rod Walsh	23/09/2004 Date

Nokia Research Center
Toni Paila

v0.055, [redacted]

ASAB Protocol Specifications

Extensions to SDP

Owner:	Toni Paila
Scope:	ASAB protocol extensions for SDP
Status:	proposal (0.055)
Document ID:	ASAB-D2
Location:	Documentum ID 4-11492

Approved by

TypeDateHere TypeNameHere

Version history

Date	Author	Version	Status[1]	Description
[redacted]	Toni Paila	0.00	Draft	Document started
[redacted]	Toni Paila	0.01	Draft	Added comments from Rod/Juha. Re-organised. Added more explanations.
[redacted]	Juha-Pekka Luoma	0.02	Draft	Added ABNF and more explanations to chapter 4, Special mappings.
[redacted]	Toni Paila	0.03	Draft	Added references, combined two edit versions (Juha's & Toni's) and streamlined <coverage> to chapter 4
[redacted]	Toni Paila, J-P Luoma	0.04	Draft	Added section 4.6
[redacted]	Juha-Pekka Luoma	0.05	Draft	Added notes on subnet mask, added references. Small changes to ABNF.
[redacted]	Toni Paila	0.051	Draft	Reviewed, added comments and corrected typos. Added timing to 5.2 and 5.3
[redacted]	Toni Paila, J-P Luoma	0.052	Draft	Reviewed
[redacted]	Toni Paila	0.053	Draft	Added ABNF and example for control channel announcements
[redacted]	Juha-Pekka Luoma	0.054	Draft	Added missing explanations to examples.
[redacted]	Toni Paila, J-P Luoma	0.055	Proposal	Modified ABNF for DVB-network type, accepted changes. Proposed specification for review.

[1]
Draft: Unfinished document representing authors' views.
Proposal: Reviewed by the project manager, represents the views of the project group.
Reviewed: Reviewed by the QA Engineer assigned to the project.
Final: Deliverable that has been formally approved by the customer of the project.

Contents

1. INTRODUCTION

This is a protocol specification extending the basic SDP. This document is result of ASAB project and, thus the requirements for extensions come from the requirements of ASAB.

The purpose of this document is to extend SDP beyond its current use in two ways. First, this means that use SDP to describe services instead of plain multicast sessions. A multicast sessions thus becomes a simple basic service. An example of another services that we want to describe is unicast connectivity. In addition, we consider several new parameters that are needed when services are announced in broadcast network.

Second, we extend the SDP to be able to express cell-level mappings. This means, that there will be special annoucements that actually describe mappings between IP addresses and cell-paramenters. Thus, these announcements do not descrbe a specific "session".

This document is structured as follows.

- In chapter 3, we give two sets of requirements that are needed, but which the current SDP does not fulfil.
- In chapter 4, we describe the extensions to the basic SDP protocol to support new session-level attributes. The specification follows the principles of SDP specification. Thus we use ABNF together with a set of examples.
- In chapter 5, we extend the SDP futher to cover the special mappings. The specification follows the principles of SDP specification. Thus we use ABNF together with a set of examples.

2. REFERENCES

[1] Session Description Protocol, RFC-2327, http://www.ietf.org/rfc/rfc2327.txt

[2] ASAB Requirements specification, ASAB-D1

[3] Describing session directories in SDP, http://www.ietf.org/intemet-drafts/draft-ietf-mmusic-sdp-directory-type-02.txt

[4] Session Announcement Protocol, RFC-2974, http://www.ietf.org/rfc/rfc2974.txt

[5] The PINT Service Protocol, RFC-2848, http://www.ietf.org/rfc/rfc2848.txt

[6] Media Gateway Control Protocol (MGCP), RFC-2705, http://www.ietf.org/rfc/rfc2705.txt

[7] Augmented BNF (ABNF) for syntax specifications, RFC-2234, http://www.ietf.org/rfc/rfc2234.txt

[8] DVB specification for Service Information (SI), ETSI EN 300 468.

[9] Guidelines on Implementation and Usage of Service Implementation, ETSI ETR 211.

3. REQUIREMENTS FOR SERVICE DESCRIPTION PROTOCOL

In this chapter we list and explain the requirements for the service description protocol. The base protocol for service descriptions was chosen to be SDP [1]. Thus this means, what kind of information we need to be able to express with SDP and how to extend it. The requirements listed in this chapter originate from ASAB requirement specification [2].

We have iterated, refined and categorised the requirements. There are two basic categories of requrements: session-level attributes and special mappings. The session-level attributes enhance the expression power of an single service announcement. The following is a set of requirements in that category

1. **What is the cost of the service?** Not all the services are freely accesible. Thus, it is necessary to be able to express the cost, cost mode and currency related to the service
2. **What is the availability and status of a service?** The basic SDP expects that the session is directly available during the active time window (between start time and stop time). However, this is too restrictive. It is necessary to be able to express that the session is available dynamically, for example via voting. Moreover, the session can be off-air or on-air
3. **How to get the service and where to get it?** It is necessary to be able to express the method of getting the service. For example, to receive an encrypted multicast service the user needs to send his authentication credentials to the network via a return channel. The user learns the IP-address of target host and the authentication mechanism through this extension

Second category of extension contains the special mappings / services. There are four kinds of service descriptions that are needed:

1. **Signaling the parameters of a DVB-T cell.** We must be able to signal the detailed link level parameters of a DVB-T cell in an SDP announcement. This helps the client as the scanning time to find the link is greatly reduced.
2. **Mapping a set of IP multicast addresses to cell ids.** Multicast sessions are carried on IP packets that have an multicast address. We must be able to define the mapping between a set of multicast addresses and the actual cell ids that contains the session. The receiver can then subsequently learn the cell-level parameters from the first type of an annoucement.
3. **What is the coverage of the service and how does it change?** It is necessary to be able to express the coverage of the service in question in terms of logical, unique cell identifiers of an access system. In addtion, it is equally important to be able to signal the end users about future changes in the coverage.
4. **Describing logical service announcement channels.** In some cases it is benificial to group the service announcements on a single logical channel. For example, in the case of DVB-T changing the reception frequency and re-tuning takes time. To help the receivers, there might be a cell that is used to annouce all the available sessions. One tuned to that cell, users receive fast all the annoucements and other mappings.
5. **Announcing unicast connectivity.** This is to enable announcing of unicast connectivity (or to describe network access service, NAS).

4. PROTOCOL EXTENSION SPECIFICATION – SESSION LEVEL ATTRIBUTES

There are two ways the following extensions in this chapter can appear. First, the new session level attributes can appear within a normal SDP description. In this case the extension attributes cannot be added, modified or removed by any third party (proxy) without breaking the SAP security checksum.

Second, the new session level attributes can appear in a separate SDP description. In this case, there will be two (or more) announcements. The first one describes the basic session/service. Then, the latter one(s) just append(s) further information to the first description. The method to link additional annoucements to the original description is to use unique <sessionid> attribute.

Using two separate annoucements is the recommended way to expand the SDP.

4.1 Cost of service

To be able to describe cost of a service, we introduce a new session-level attribute.

```
a=cost:<cost>
```

4.1.1 The ABNF for <cost>

```
<cost>              = cost-amount space
                      cost-unit space
                      cost-rate space
                      cost-type

cost-amount         = INTEGER

cost-unit           = "USD" / "EUR"

cost-rate           = INTEGER
                    ; in case of time based billing
                    ; this is the interval in seconds

cost-type           = "time" / "size" / "one-off"
                    ; default "one-off"
```

4.1.2 Example: Multicast session having time-based fee (using one annoucement)

The following announcement describes a normal multicast session. In addition, cost-parameter tells that the cost of receiving the session is EUR 1 per hour.

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.123.1.120/127
t=2873397496 2873404696
a=recvonly
a=cost:1 EUR 3600 time  -- 1 EUR/hour
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

4.1.3 Example: Multicast session having time-based fee (using separate annoucements)

This example describes exactly the same information as 4.1.2, but now using two separate announcements. This is recommended way.

<u>Announcement 1 – plain service description</u>

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.123.1.120/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

<u>Announcement 2 – additional cost information</u>

```
v=0
o=operator 123456789 123456789 IN IP4 111.122.133.144
s=Cost information
i=Additional cost information
t=2873397496 2873404696
a=sessionid: mhandley 2890844526 IN IP4 126.16.64.4
a=cost:1/EUR/3600/time  -- 1 EUR/hour
```

4.2 Availability of service (dynamic/static) and contact information

Active time for a service is the time between start time and stop time as expressed by the t-field of an service description. Withing this active time, the service can be available in three ways. First, the session can be available statically. This is the default case in basic SDP. Second, the service can be available dynamically on-air or off-air. The service being on-air dynamically means that the service is currently available. However, the service may go off-air withing the active time. Last, the service can be available dynamically, off-air. This means that the service exists, but is not on-air currently. For example services that need to be voted or are based on popularity can first exist as dynamically off-air. When the service actually becomes available on-air, the service state changes to dymically avaliable, on-air.

To meet the requirements of expressing the service availablity, we introduce a new session level attribute

```
a=available:<available>
```

When a service is announced as dynamically available, but currently off-air, we often need to specify how to get the service and where to contact. A new session-level attribute serves the purpose.

```
a=contact:<contact>
```

4.2.1 The ABNF for <available>

```
<available>             = "static"
                          / "dynamic/on-air"
                          / "dynamic/off-air"
                        ; if this attribute is not present,
                        ; default by RFC-2327 is "static"
                        ; in case of dynamic/on-air or dynamic/off-air,
                        ; consult "a=contact"
```

4.2.2 The ABNF for <contact> is:

```
<contact>               = contact-address "/" contact-type
contact-address         = IP4-address
                        ; "Where to get"
                        ; -- Address of attendant / server / join listener
                        ; where to send joins, etc

contact-type            = protocol
                        ; "How to get"

protocol                = "IGMP-JOIN" / "IGMP-VOTE" / "NWAP"
```

4.2.3 Example: Session available dynamically, currently off-air (using one annoucement)

The following announcement describes a normal multicast session, which is offered to the end users. However, the session is not yet on-air (it's state is dynamic/off-air). Users can vote for the session to become on-air by sending a special IGMP-VOTE message to a host attendant.ucl.ac.uk.

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.123.1.120/127
t=2873397496 2873404696
a=recvonly
a=available:dynamic/off-air
            -- see the <contact> parameter to see more
a=contact:attendant.ucl.ac.uk/IGMP-VOTE
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

4.2.4 Example: Session available dynamically, currently off-air (using separate announcements)

This example describes exactly the same information as 4.2.3, but now using two separate announcements. This is recommended way.

<u>Announcement 1 – plain service description</u>

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.123.1.120/127
t=2873397496 2873404696
```

```
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

Announcement 2 – additional availability information

```
v=0
o=operator 123456789 123456789 IN IP4 111.122.133.144
s=Cost information
i=Additional cost information
t=2873397496 2873404696
a=sessionid: mhandley 2890844526 IN IP4 126.16.64.4
a=available:dynamic/off-air
            -- see the <contact> parameter to see more
a=contact:attendant.ucl.ac.uk/IGMP-VOTE
```

5. PROTOCOL EXTENSION SPECIFICATION – SPECIAL MAPPINGS

5.1 DVB-T cell parameter announcement

This announcement describes the physical transport parameters of a single DVB-T broadcast cell. The following ABNF extends the SDP syntax defined in RFC 2327. In addition, new session- and media-level attributes described below are defined for the announcement.

The complete DVB-T cell parameter announcement thus consists of two parts: connection parameter and cell-describing attributes as session-level or media-level parameters.

```
c=<nettype> <addrtype> <connection-address>
<cell-attribute-fields>
<cell-media-fields>
```

5.1.1 The ABNF for <nettype>, <addrtype> and <connection address>

```
<nettype>               = "DVB/CELL"

<addrtype>              = = "IP4" / "SI"
                        ; IP4: IPv4 address identifies the cell
                        ; SI: cell_id in DVB Service Information
                        ; identifies the cell
                        ; note - should be extended later to support IPv6 as well


<connection-address>    = dvb-cell-addr


dvb-cell-addr           = dvb-cell-ip4-addr / dvb-cell-si-addr

dvb-cell-ip4-addr       = unicast-addr
                        ; unicast-addr defined in RFC 2327
                        ; note - private IP addresses should NOT be used
                        ; as cell IDs

dvb-cell-si-addr        = original-network-id "/" dvb-cell-id

original-network-id     = decimal-ushort
                        ; original_network_id defined in DVB SI

dvb-cell-id             = decimal-ushort
                        ; cell_id defined in DVB SI

decimal-ushort          = 1*(DIGIT)
                        ; unsigned 16-bit integer
```

5.1.2 The ABNF for <cell-attribute-fields>

```
<cell-attribute-fields> = normal-cell-attribute-fields
                          / abbreviated-cell-attribute-fields
                        ; based on <attribute-fields> defined in RFC 2327
                        ; mandatory sub-fields indicated in comments below

normal-cell-attribute-fields = *("a=" normal-cell-attribute CRLF)

abbreviated-cell-attribute-fields = *("a=" abbreviated-cell-attribute CRLF)

normal-cell-attribute   = bandwidth
                          / FFT-mode
                          / constellation
```

```
                                    / coderate
                                    / guard-interval
                                    / hierarchy
                                    / hierarchical-priority
                                    / dvb-framing
                                    / normal-bearer
                                  ; these DVB-T parameters defined in the
                                  ; DVB SI specification

abbreviated-cell-attribute = dvb-framing
                                      / abbreviated-bearer

bandwidth                         = "dvb-t-bandwidth:" bandwidth-attribute
                                  ; mandatory

fft-mode                          = "dvb-t-fft:" fft-mode-attribute
                                  ; mandatory

constellation                     = "dvb-t-constellation:" constellation-attribute
                                  ; mandatory

coderate                          = "dvb-t-coderate:" coderate-attribute
                                  ; mandatory

guard-interval                    = "dvb-t-guard-interval:" guard-interval-attribute
                                  ; mandatory

hierarchy                         = "dvb-t-hierarchy:" hierarchy-attribute
                                  ; mandatory

hierarchical-priority             = "dvb-t-hierarchical-priority:"
                                  hierarchical-priority-attribute
                                  ; ignored if hierarchy == "none"
                                  ; mandatory if hierarchy != "none"

dvb-framing                       = "framing:" dvb-framing-mode
                                  ; mandatory

dvb-framing-mode                  = "dvb/mpe"
                                  ; DVB multiprotocol encapsulation,
                                  ; other alternatives could be added here

normal-bearer                     = "bearer:" normal-dvb-bearer
                                  ; mandatory

abbreviated-bearer                = "bearer:" abbreviated-dvb-bearer
                                  ; mandatory

normal-dvb-bearer                 = "dvb-t"

abbreviated-dvb-bearer            = "dvb-t"
                                    space bandwidth-attribute
                                    space fft-mode-attribute
                                    space constellation-attribute
                                    space coderate-attribute
                                    space guard-interval-attribute
                                    space hierarchy-attribute
                                    [ space hierarchical-priority-attribute ]

bandwidth-attribute               = "7" / "8"
                                  ; bandwidth in MHz


fft-mode-attribute                = "2" / "8"
                                  ; FFT mode used, 2k or 8k
```

```
constellation-attribute = "QPSK" / "16QAM" / "64QAM"

coderate-attribute      = "1/2" / "2/3" / "3/4" / "5/6" / "7/8"

guard-interval-attribute= "1/32" / "1/16" / "1/8" / "1/4"

hierarchy-attribute     = "none" / "1" / "2" / "4"
                        ; alpha value for
                        ; hierarchical coding, or "none" if
                        ; hierarchical coding is not used

hierarchical-priority-attribute     = "low" / "high"
```

5.1.3 ABNF for <cell-media-fields>

```
<cell-media-fields>     = 1*(cell-media-field)

cell-media-field        = "m=nas/none" CRLF cell-media-attributes

cell-media-attributes   = 1*("a=" cell-subcell-attribute CRLF)

cell-subcell-attribute  = "subcell:" [ dvb-subcell-id space ]
                          cell-central-frequency-attribute
                          [ "/" cell-coverage-attribute ]

dvb-subcell-id          = decimal-uchar
                        ; decimal-uchar defined in RFC 2327
                        ; subcell_id defined in DVB SI

cell-central-frequency-attribute = float
                        ; central frequency in MHz

cell-coverage-attribute = "coverage:" center-coord "/" radius

float                   = *(DIGIT) "." 1*(DIGIT)

center-coord            = float ("N" / "S") "/" float ("E" / "W")
                        ; in degrees, latitude and longitude

radius                  = float "/" float
                        ; in kilometres, north-south and east-west
```

5.1.4 Example: Mapping of (sub)cell ids – normal notation

The following announcement introduces a DVB cell, identified by the IP address 15.21.12.34. The session-level attribute fields with the prefix "dvb-t-" descibe DVB-T link level parameters, common to all subcells of a DVB-T cell. Last, two media sections in the end identify two subcell s with subcell IDs 1 and 2, each with different geographical coverage.

```
v=0
o=- 430593082 232365346 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Parameters of a DVB-T cell and subcells
c=DVB/CELL IP4 15.21.12.34
a=dvb-t-bandwidth:8
a=dvb-t-fft:8
a=dvb-t-constellation:16QAM
a=dvb-t-coderate:2/3
a=dvb-t-guard-interval:1/8
a=dvb-t-hierarchy:none
a=dvb-t-hierarchical-priority:high
a=framing:dvb/mpe
```

```
a=bearer:dvb-t
t=4983200 1274217
m=nas/none
a=subcell:1 450.2/60.31N/12.44E/3.1/2.5
m=nas/none
a=subcell:2 516.3/60.28N/12.45E/2.9/2.8
```

5.1.5 Example: Mapping of (sub)cell ids – abbreviated notation

The following announcement descibes the DVB-cell in a shorter, abbreviated notation. This notation bundles all the link-level parameters of a DVB-T cell into one `bearer` attribute field.

In this case, the cell is identified by the IP address 15.21.12.35 and contains just one subcell (a DVB-T cell always consists of at least one subcell in the ASAB announcement syntax). The optional subcell ID parameter has been omitted from the `subcell` field, as there is no need to differentiate between the subcells of this cell.

```
v=0
o=- 430593082 232365346 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Parameters of a DVB-T cell and subcells
c=DVB/CELL IP4 15.21.12.35
a=framing:dvb/mpe
a=bearer:dvb-t 8 8 16QAM 2/3 1/8 none
t=4983200 1274217
m=nas/none
a=subcell:491.23/23.59S/90.63W/40.9/38.5
```

5.1.6 Example: Mapping of (sub)cell Ids – abbreviated notation

This announcement describes the paramters of a DVB-T cell consisting of three subcells. Because the hierarchy-attribute subfield contains a value other than "none", the hierarchical-priority-attribute subfield must be included, and has a value of "high" in this case. The inclusion of subcell coverage parameters is recommended, although not mandatory - in this example the coverage parameters have been omitted.

```
v=0
o=- 430593082 232365346 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Parameters of a DVB-T cell and subcells
c=DVB/CELL IP4 15.21.12.36
a=framing:dvb/mpe
a=bearer:dvb-t 8 8 64QAM 1/2 1/16 2 high
t=4983200 1274217
m=nas/none
a=subcell:1 460.1
m=nas/none
a=subcell:2 510.5
m=nas/none
a=subcell:3 570.9
```

5.2 Mapping from DVB-T cells to sessions

This announcement identifies one or more DVB broadcast cells, and for each cell describes the group of sessions being transmitted in that cell. New media-level attributes described below are defined for the DVB-T cell to session mapping announcements. In addition to the mapping announcements defined here, clients also need to receive normal SAP announcements describing the sessions.

Similarly to DVB-T cell parameter announcements described earlier, cell-to-session announcements consist of two parts: a `<dvb-network-connection-field>` identifying the DVB network, and media-level attributes defined as `<cell-mapping-media-fields>`.

The following subsections extend the SDP syntax defined in RFC 2327. The ABNF entries `connection-field` and `media-descriptions` defined in RFC 2327 are replaced by `<dvb-network-connection-field>` and `<cell-mapping-media-fields>`, respectively.

5.2.1 The ABNF for <network-connection-field>

```
<dvb-network-connection-field> = "c=" network-nettype
                                 space network-addrtype
                                 space network-connection-address CRLF

network-nettype          = "DVB/NETWORK"

network-addrtype         = "SI"

network-connection-address = dvb-network-addr


dvb-network-addr         = original-network-id

original-network-id      = decimal-ushort
                         ; original_network_id defined in DVB SI

decimal-ushort           = 1*(DIGIT)
                         ; unsigned 16-bit integer
```

5.2.2 The ABNF for media-level attributes

```
<cell-mapping-media-fields> = 1*("m=nas/none" CRLF
                              cell-connection-field CRLF
                              cell-mapping-attribute-fields)

cell-connection-field    = "c=" nettype space cell-c-addrtype-addr

cell-mapping-attribute-fields = 1*("a=" unique-session-id-attribute CRLF)

cell-nettype             = "DVB"

cell-c-addrtype-addr     = "CELL" dvb-cell-addrtype dvb-cell-addr

unique-session-id-attribute = "sessionid:" username space sess-id space
                              nettype space addrtype space addr [":" timing]
                         ; globally unique session identifier that maps to
                         ; the "c=" field of an RFC 2327 session announcement

dvb-cell-addrtype        = "IP4" / "SI"

dvb-cell-addr            = dvb-cell-ip4-addr / dvb-cell-si-addr

username                 = safe
                         ; username and safe defined in RFC 2327

sess-id                  = 1*(DIGIT)
                         ; sess-id defined in RFC 2327
                         ; should be unique for this originating username/host

addrtype                 = "IP4" / "IP6"
                         ; addrtype defined in RFC 2327
```

```
addr                   = FQDN / unicast-address
                       ; addr, FQDN and unicast-address defined in RFC 2327

dvb-cell-ip4-addr      = unicast-addr
                       ; unicast-addr defined in RFC 2327

dvb-cell-si-addr       = original-network-id "/" dvb-cell-id

original-network-id    = decimal-ushort
                       ; original_network_id defined in DVB SI

dvb-cell-id            = decimal-ushort
                       ; cell_id defined in DVB SI

decimal-ushort         = 1*(DIGIT)
                       ; unsigned 16-bit integer
```

5.2.3 ABNF for <timing>

```
<timing>               = start-time stop-time
                       ; the time this attribute becomes active
                       ; - If timing does not exist, assume session
                       ;   level t-attribute
                       ; - If start-time == 0, assume session level start-time
                       ; - If stop-time == 0, assume session level stop-time
```

5.2.4 Example: DVB-T cell to session mappings

The following announcement declares that the DVB cell identified by IP address 15.21.12.34 contains three multicast sessions and the cell identified by IP addrss 15.21.12.35 contains four multicast sessions. The contained sessions are refereed with sessionid-attribute. Note that both cells contain the session (a=sessionid:- 3398739487 IN IP4 136.34.12.2). Note also that the session (a=sessionid:- 3398739487 IN IP4 136.34.123.26) Migrates from cell 15.21.12.34 to 15.21.12.35. The transition takes palce during time interval 344430000...344450000.

```
v=0
o=- 3374573294 3399187242 IN IP4 datacast.digita.fi
s=Network Access Service (NAS) announcement
i=Mapping from DVB-T cell(s) to sessions
c=DVB/NETWORK SI 32765
t=3398713874 0
m=nas/none
c=DVB/CELL IP4 15.21.12.34
a=sessionid:- 3398737481 IN IP4 136.34.12.2
a=sessionid:- 3398739487 IN IP4 136.34.123.26 0 344450000
a=sessionid:- 3398983458 IN IP4 mediacast.sonera.fi
a=sessionid:admin 3398778932 IN IP4 138.34.253.9
m=nas/none
c=DVB/CELL IP4 15.21.12.35
a=sessionid:- 3398737481 IN IP4 136.34.12.2
a=sessionid:- 3398739487 IN IP4 136.34.123.26 344430000 0
a=sessionid:- 3398983453 IN IP4 mediacast.sonera.fi
a=sessionid:root 3398798446 IN IP4 139.43.56.76
a=sessionid:demo 3398773348 IN IP4 mediacast.sonera.fi
```

5.3 Mapping from sessions to DVB-T cells

It is necessary to be able to define the coverage of a service in terms of logical scope. We define the scope in terms of a list of cells. In addition, it is necessary to describe such actions as coverage expansion, coverage contraction and migration of service from a cell to another. This section explains how to achieve all this.

We intruduce a new announcement that identifies one or more sessions, and for each session describes the group of DVB-T cells where the session is being transmitted. New media-level attributes described below are defined for session to DVB-T cell mapping announcements. In addition to the mapping announcements defined here, clients also need to receive normal SAP announcements describing the sessions.

Similarly to DVB-T cell parameter announcements described earlier, cell-to-session announcements consist of two parts: an optional `<dvb-network-connection-field>` (defined in 5.2.1) identifying the DVB network, and media-level attributes defined here as `<sess-mapping-media-fields>`.

The following ABNF extends the SDP syntax defined in RFC 2327. The ABNF entries `connection-field` and `media-descriptions` defined in RFC 2327 are extended by `<dvb-network-connection-field>` and `<sess-mapping-media-fields>`, respectively.

5.3.1 The ABNF for media-level attributes

```
<sess-mapping-media-fields> = 1*("m=nas/none" CRLF
                              "a=" unique-session-id-attribute CRLF
                              sess-mapping-attribute-fields)
                              ; unique-session-id-attribute defined earlier

sess-mapping-attribute-fields = 1*("a=" cell-id-attribute CRLF)

cell-id-attribute         = "cellid:" dvb-cell-addrtype space dvb-cell-addr
                            ["/" cell-id-range] [":" timing]
                          ; dvb-cell-addrtype and dvb-cell-addr defined earlier

cell-id-range             = POS-DIGIT *(DIGIT)
                          ; number of consecutive cell ids - if no
                          ; cell-id-range is given, is assumed
```

5.3.2 Example: session to DVB-T cell mappings

The following announcement is an example of DVB-T cell mappings. All media descriptions in cell mappings start with the "m=nas/none" m-field[2]. In each media description, a session identided by the `sessionid` field is mapped to one or more DVB-T cells identified by `cellid` fields. Because multiple media descriptions can be included in an announcement (as with standard SDP [1]), this announcement format allows any number of sessions to be mapped to the cells delivering those sessions.

The inclusion of a c-field with the *network_identifier* [8], [9] of a DVB network is recommended but not mandatory.

[2] The syntax of this form of the m-field originatesfrom the Media Gateway Control Protocol specification [6]

```
v=0
o=- 3374573295 3399107243 IN IP4 datacast.digita.fi
s=Network Access Service (NAS) announcement
i=Mapping from session(s) to DVB-T cells
c=DVB/NETWORK SI 32765
t=3398715874 0
m=nas/none
a=sessionid:- 3398739487 IN IP4 136.34.123.26
a=cellid:IP4 15.21.12.34
a=cellid:IP4 15.21.12.35
m=nas/none
a=sessionid:- 3398983458 IN IP4 mediacast.sonera.fi
a=cellid:IP4 15.21.12.34
a=cellid:IP4 15.21.12.35
m=nas/none
a=sessionid:admin 3398778932 IN IP4 138.34.253.9
a=cellid:IP4 15.21.12.34
m=nas/none
a=sessionid:root 3398798446 IN IP4 139.43.56.76
a=cellid:IP4 15.21.12.35
m=nas/none
a=sessionid:demo 3398773348 IN IP4 mediacast.sonera.fi
a=cellid:IP4 15.21.12.35
```

5.4 Describing logical service announcement channels

In this case we want to announce the presence of a logical channel (for example a cell) that further contains service announcements. We achieve this with a new individual SDP description. The method is modified from a proposed way of describing session derectories with SDP [3].

The session-level connection parameter (c-field) specifies the actual multicast address of the announcement channel. Then, session-level attribute a=cellid uniquely defines the cell, In which the above mentioned multicast address is available. Media parameters are set os descrbed in [3]. Last, media-level attribute a=cellid can be used to specify to which cells the announcements belong.

5.4.1 ABNF for a logical service announcement channel description

```
<logical service announcement channel description>
                           = v-field
                             o-field
                             s-field
                             c-field
                             *(cell-reference)
                             *(sdr-media-bundle)

cell-reference             = cell-id-attribute

sdr-media-bundle           = sdr-media-format
                             *(cell-reference)
```

5.4.2 Example: announcing neighbour control channel of neigbouring cells

The following description presents an example how a cell can announce the control channels of it neigbour cells. These channels then in tum contain majority of announcements related to the espective cell.

```
v=0
```

```
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=Logical control channel announcement
c=IN IP4 224.2.127.252/255 -- LBA specific cell-local-ann. multicast group
a=cellid:IP4 15.21.12.22 -- Where is the mcast address of c-field available
t=0 0
m=application 9875 SAP SDP
a=cellid:IP4 15.21.12.22     -- Target cell #1 to be described
a=cellid:IP4 15.21.12.23     -- Target cell #2 to be described
a=cellid:IP4 15.21.12.24     -- Target cell #3 to be described
```

5.4.3 Example: describing control channel

The following presents an example of how to announce logical announcement channels with SDP. The service description below expresses that the IP address 224.2.127.252 carries a control channel for cells 15.21.12.34, 15.21.12.31 and 15.21.12.32. For the first of these the description protocol is SDP and for the two last ones the description protocol is SDL-XML. The identical service announcement channel carrying 224.2.127.252 is available in cells 15.21.12.22 and 15.21.12.26.

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=Logical control channel announcement
c=IN IP4 224.2.127.252/255 -- LBA specific cell-local-ann. multicast group
a=cellid:IP4 15.21.12.22 -- Where is the mcast address of c-field available
a=cellid:IP4 15.21.12.26 -- Additional cell, where the mc addr is available
t=0 0
m=application 9875 SAP SDP
a=cellid:IP4 15.21.12.34     -- Target cell #1 to be described
m=application 9876 SAP SDP-XML
a=cellid:IP4 15.21.12.31     -- Target cell #2 to be described
a=cellid:IP4 15.21.12.32     -- Target cell #3 to be described
```

5.5 Mapping DVB service components to IP addresses

This announcement defines the mapping of DVB service components to IP addresses within a single DVB-T broadcast cell. These announcements only need to be transmitted in cells carrying IP data on more than one PID.

The DVB service component to IP address mapping consists of two parts: a <dvb-cell-connection-field> identifying a DVB cell, and media-level attributes defined as <comp-mapping-media-fields>.

The following subsections extend the SDP syntax defined in RFC 2327. The ABNF entries connection-field and media-descriptions defined in RFC 2327 are replaced by <dvb-cell-connection-field> and <comp-mapping-media-fields>, respectively.

Notice the slightly non-standard use of the subnet mask length in the subnet-attribute field defined below. In ASAB announcement protocol, the subnet mask length is used to indicate individual IP addresses or address ranges, for both unicast and multicast IP addresses. If the subnet mask length is smaller than the length (in bits) of its associated IP address, a range of IP addresses is described. However, if the subnet mask has the same length as the IP address, an individual IP address is indicated.

For example:

```
subnet:224.0.0.0/16      -- A subsection of the multicast IP address
                          -- range (224.0.0.0 - 224.0.255.255)

subnet:236.25.243.190/32 -- An individual multicast IP address (236.25.243.190)
```

5.5.1 The ABNF for <cell-connection-field>

```
<dvb-cell-connection-field> = "c=" cell-nettype
                              space cell-addrtype
                              space cell-connection-address CRLF

cell-nettype            = "DVB"

cell-addrtype           = "CELL"

cell-connection-address = dvb-cell-addrtype space dvb-cell-addr
                        ; dvb-cell-addrtype and dvb-cell-addr defined earlier
```

5.5.2 The ABNF for <comp-mapping-media-fields>

```
<comp-mapping-media-fields> = 1*("m=nas/none" CRLF
                            comp-connection-field CRLF
                            comp-mapping-attribute-fields)

comp-connection-field   = "c=" comp-c-nettype-addrtype space
                        service-component

comp-c-nettype-addrtype = "DVB/SERVICE"

comp-mapping-attribute-fields = 1*("a=" subnet-attribute CRLF)

subnet-attribute        = "subnet:" subnet-nettype space subnet-addrtype space
                          subnet-addr "/" subnet-mask-length
                        ; describes an address range
                        ; extends the definition from RFC 2705
                        ; note - subnets being transmitted within the same
                        ; DVB-T cell must not overlap

subnet-nettype          = "IN"

subnet-addrtype         = "IP4" / "IP6"

subnet-addr             = unicast-address / multicast-address
                        ; unicast-address and multicast-address defined
                        ; in RFC 2327

subnet-mask-length      = decimal-uchar
                        ; length (in bits) of the subnet mask
                        ; decimal-uchar defined in RFC 2327

service-component       = service-locator ["/" component-tag]
                        ; component-tag can be left out if the service
                        ; contains only one data broadcast component carrying
                        ; IP over MPE

subnet                  = "IN" space
                          subnet-addrtype space
                          subnet-addr "/" subnet-mask-length

service-locator         = service-path / textual-service-identifier
```

```
component-tag             = decimal-uchar

service-path              = original-network-id "/"
                            service-id
                          ; original-network-id defined earlier

textual-service-identifier = FQDN
                          ; FQDN (fully-qualified domain name) defined in RFC 2327

service-id                = decimal-ushort
                          ; decimal-ushort defined earlier
                          ; service_id defined in DVB SI
```

5.5.3 Example: Mapping IP addresses to DVB service components

The example below shows the mapping of the entire multicast IPv4 address range (224.0.0.0 - 239.255.255.255) to four components of a DVB service being broadcast in a DVB-T cell. The cell is identified by a c-field in the session-level part of the announcement. The mapping from IP address ranges to DVB service components is defined in the media-level part of the announcement, consisting of one or more media descriptions. Each media description includes a c-field identifying a DVB service and a component within that service. The media description further contains one or more subnet fields that describe the address ranges being transmitted on the given service component.

In this example, the selection between the four components takes place according to the two most significant bits of a multicast IPv4 address, following the four-bit address prefix that is constant for all multicast IPv4 addresses. Thus, a subnet mask length of 6 bits is used in the subnet fields below.

```
v=0
o=- 458298724 582732834 IN IP4 datacast.sonera.fi
s=Network Access Service (NAS) announcement
i=IP address to DVB service component mappings
c=DVB/CELL IP4 131.228.7.11
m=nas/none
c=DVB/SERVICE multicast.medianet.sonera.fi/1
a=subnet:IN IP4 224.0.0.0/6
m=nas/none
c=DVB/SERVICE multicast.medianet.sonera.fi/2
a=subnet:IN IP4 228.0.0.0/6
m=nas/none
c=DVB/SERVICE multicast.medianet.sonera.fi/3
a=subnet:IN IP4 232.0.0.0/6
m=nas/none
c=DVB/SERVICE multicast.medianet.sonera.fi/4
a=subnet:IN IP4 236.0.0.0/6
```

5.6 Access mappings

Access mappings enable clients to obtain a return data path to a media operator that offers a datacasting service via DVB-T. Clients can then be provided with a "hybrid network" connection to the Internet, where the forward data path is provided via DVB-T and the return data path via some other network. The availability of a return data path enables clients to use unicast protocols and/or participate in "voting" for the selection of dynamic multicast content on the DVB-T network.

By sending access mappings, datacast operators can provide clients the IP address of a login server or the phone number of a modem pool. Providing an IP address is preferable, as clients are then free to use any Internet Service Provider (ISP) for connecting to the media operator. General-purpose ISP phone numbers may also be advertised using access mappings, without requiring the client to use a particular ISP.

Access mappings consist of standard SDP session and time attributes, followed by the media-level attributes defined in the following as <access-media-fields>.

5.6.1 The ABNF for access mappings

```
<access-media-fields>   = 1*(access-media-field)

access-media-field      = "m=" access-nas-field CRLF access-connection-field CRLF
                          access-attribute-fields CRLF

access-nas-field        = "nas/" nas-authentication

access-connection-field = "c=" c-access-nettype space c-access-addrtype space
                          c-access-connection-address CRLF

access-attribute-fields = 1*("a=" access-attribute-field CRLF)
                        ; mandatory access attributes indicated below

nas-authentication      = "none" / "login" / "chap" / "pap" / "ipsec" / "l2tp"
                        ; these authentication methods defined in RFC 2705

c-access-nettype        = "IN" / "TN"
                        ; TN for Telephone Network, as in RFC 2848

c-access-addrtype       = in-addrtype / tn-addrtype

c-access-connection-address = in-connection-address / tn-connection-address

access-attribute-field  = framing-attribute
                          / bearer-attribute
                          / cell-id-attribute
                          / subnet-attribute
                        ; mandatory cell-id-attribute defined earlier
                        ; subnet-attribute defined earlier

in-addrtype             = "IP4" / "IP6"

tn-addrtype             =  "RFC2543"
                        ; this address type defined in RFC 2848

in-connection-address   = FQDN / unicast-addr
                        ; FQDN and unicast-addr defined in RFC 2327

tn-connection-address   = inp-addr / ldp-addr

framing-attribute       = "framing:" framing-mode
                        ; indicates the layer 2 framing used
                        ; extends the definition from RFC 2705

bearer-attribute        = "bearer:" bearer-type
                        ; extends the definition from RFC 2705

inp-addr                = " " POS-DIGIT 0*(("-" DIGIT) / DIGIT)
                        ; global phone number,
                        ; defined as INPAddr in RFC 2848
```

```
ldp-addr                    = DIGIT 0*(("-" DIGIT) / DIGIT)
                            ; local phone number,
                            ; defined as LDPAddr in RFC 2848

framing-mode                = dvb-framing-mode /
                              "ppp-sync" /
                              "ppp-asynch" /
                              "ppp-hdlc" /
                              "slip" /
                              "asynch"
                            ; dvb-framing-mode defined earlier

bearer-type                 = normal-dvb-bearer / modem-bearer / isdn-bearer
                            ; normal-dvb-bearer defined earlier

modem-bearer                = modem-standard ["/" modem-manufacturer]

isdn-bearer                 = "isdn" *(DIGIT) ["/" isdn-standard]
                            ; example values: "isdn56", "isdn64",
                            ; "isdn64/v.110", "isdn64/v.120"

modem-standard              = ("v." 1*(alpha-numeric)) /
                              "x2" /
                              "k56flex"
                            ; example values: "v.32", "v.34", "v.90", "x2", "k56flex"

modem-manufacturer          = 1*(safe)
                            ; example values: "3com", "rockwell"

isdn-standard               = 1*(safe)
                            ; example values: "v.110", "v.120"
```

5.6.2 Example: Announcing a return data path to a media network operator via a modem pool

The following example announces two dial-in modem numbers that clients can use to obtain a return data path to a media operator. The cell id(s) associated with each phone number indicate the recommended dial-in number for clients, based on the location of each client.

Each phone number is announced within a media description that starts with an m-field. The m-field is of the format "m=nas/xxx" identifying an authentication method for a Network Access Service, as defined in [6]. In this example, the authentication method "login" indicates that end-users will be prompted for a username and password for authentication.

Each media description contains a c-field describing a connection address for the return data path, given in this example as the phone number of a modem pool. Additional parameters describing the modem type (bearer attribute [6]) and link-layer framing (framing attribute [6]). Finally, one or more cells are identified (using the cellid attribute) as a target area where each return data path connection address should be used - for example to provide a local PSTN number to end-users where possible. Note that the same cell id can be listed in more than one media description within an access mapping.

```
v=0
o=- 346232972 928002543 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Subnet for DVB MPE encapsulated IP data
m=nas/login
c=TN RFC2543 +358-2-2340982
a=framing:ppp-asynch
a=bearer:v.90
```

```
a=cellid:IP4 160.237.53.1/8
m=nas/login
c=TN RFC2543 +358-3-5837272
a=framing:ppp-asynch
a=bearer:v.90
a=cellid:IP4 160.238.45.1/4
a=cellid:IP4 160.238.46.1/4
```

5.6.3 Example: Announcing a return data path to a media network operator via Internet

Similar to the the previous example, the following announcement describes the connection addresses for obtaining a return data path to a media operator. As in the above example, a number of cells is mapped to each connection address. The difference to the preceeding example is that each connection address here is given as an IP address. This allows clients to use their most preferred Internet Service Provider to connect to the indicated IP address.

```
v=0
o=- 346232972 928002543 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Subnet for DVB MPE encapsulated IP data
m=nas/login
c=IN IP4 portal1.mediacast.sonera.fi
a=cellid:IP4 160.237.53.1/8
m=nas/login
c=IN IP4 portal2.mediacast.sonera.fi
a=cellid:IP4 160.238.45.1/4
a=cellid:IP4 160.238.46.1/4
```

5.6.4 Example: Announcing connectivity to an Internet service provider via a modem pool

While the earlier announcement leaves clients with the freedom of choosing an ISP, a client may not always be configured with the current ISP connectivity details. This announcement informs clients about available ISP dial-in numbers. One or more Cell ids are optionally associated with each phone number, to indicate the recommended ISPs for clients depending on their location.

```
v=0
o=- 346232972 928002543 IN IP4 131.228.32.59
s=Network Access Service (NAS) announcement
i=Internet Service Provider connectivity info
m=nas/login
c=TN RCF2324 +358-9-6524827
a=framing:ppp-asynch
a=bearer:v.90
a=cellid:IP4 160.237.53.1/8
m=nas/login
c=TN RCF2324 +358-3-3157161
a=framing:ppp-asynch
a=bearer:v.90
a=cellid:IP4 160.238.45.1/4
m=nas/login
c=TN RCF2324 +358-2-4182732
a=framing:ppp-asynch
a=bearer:v.90
a=cellid:IP4 160.238.46.1/4
```

Patent-Agency Banner-Witcoff (EXT-RES/Washington)

From: Paila Toni (Nokia-NVO/Helsinki) **Sent:** Wed 6/6/2001 4:21 AM
To: Luoma Juha-Pekka (Nokia-NRC/Tampere)
Cc: Jalonen Esa (Nokia-NVO/Helsinki); Saarikivi Tuomo (Nokia-NVO/Helsinki); Poikela Jani (Nokia-NRC/Helsinki); Jalonen Erkka (NRC/Helsinki)
Subject: ASAB protocols v0.055 (PROPOSAL)
Attachments: ASAB-D2-Protocols_055.doc(162KB)

Moi Pekka,

Liitteenä speksi. Laittaisitko documentumiin.

-Toni

ASAB Server side specifications

Functional, technical and test specification

Owner:	Toni Paila
Scope:	ASAB server
Status:	draft
Document ID:	ASAB-D3
Location:	

Approved by

xx.yy.2001 n.n.

Version history

Date	Author	Version	Status[1]	Description
[redacted]	Toni Paila	0.00	Draft	Document started
[redacted]	Toni Paila	0.01	Draft	Additions after 1st ASAB technical meeting
[redacted]	Toni Paila	0.02	Draft	Added DB descriptions
[redacted]	Toni Paila	0.021	Draft	Extended structure to cover the whole D3. Now covers FS, TS, implementation and test parts
[redacted]	Toni Paila	0.03	Draft	Corrected E/R-diagram + modifications after received comments from UBA team in 2nd ASAB technical meeting
[redacted]	Toni Paila	0.04	Draft	Cell broker becomes a function; added several updated figures
[redacted]	Toni Paila	0.041	Draft	Shifted asab_config.queue_listen to asab_control.multicast_listen.
[redacted]	Toni Paila	0.042	Draft	Updated database tables
[redacted]	Jani Poikela	0.043	Draft	asab_control database: server_id changed to ip and port. asab_mapping: dvb_param and subcell separated.
[redacted]	Toni Paila	0.044	Draft	Shared document parts to the team. Basic structure for Management Interface chapter.
[redacted]	Erkka Jalonen	0.045	Draft	Updated database tables. Added management interface section.
[redacted]	Paila Toni	0.046	Draft	More database updates, error corrections.

[1]

Draft:	Unfinished document representing authors' views.
Proposal:	Reviewed by the project manager, represents the views of the project group.
Reviewed:	Reviewed by the QA Engineer assigned to the project.
Final:	Deliverable that has been formally approved by the customer of the project.

Contents

1. INTRODUCTION

This is the collection of specifications for the announcement server to be build in the project ASAB (Advanced Service Announcement for Broadcasting).

This specification is divided into three parts. In the first part we the functional (or architecture) specification of the server. This includes background and context for the system. In addition, we describe the system giving an overview and explaining the functionalities to be implemented in the announcer.

The second part is the server technical (or component/implementation) specification of tryhe server.

The last part serves as test specification.

The common terms of reference used in the project ASAB are listed in the Annex A.

2. REFERENCES

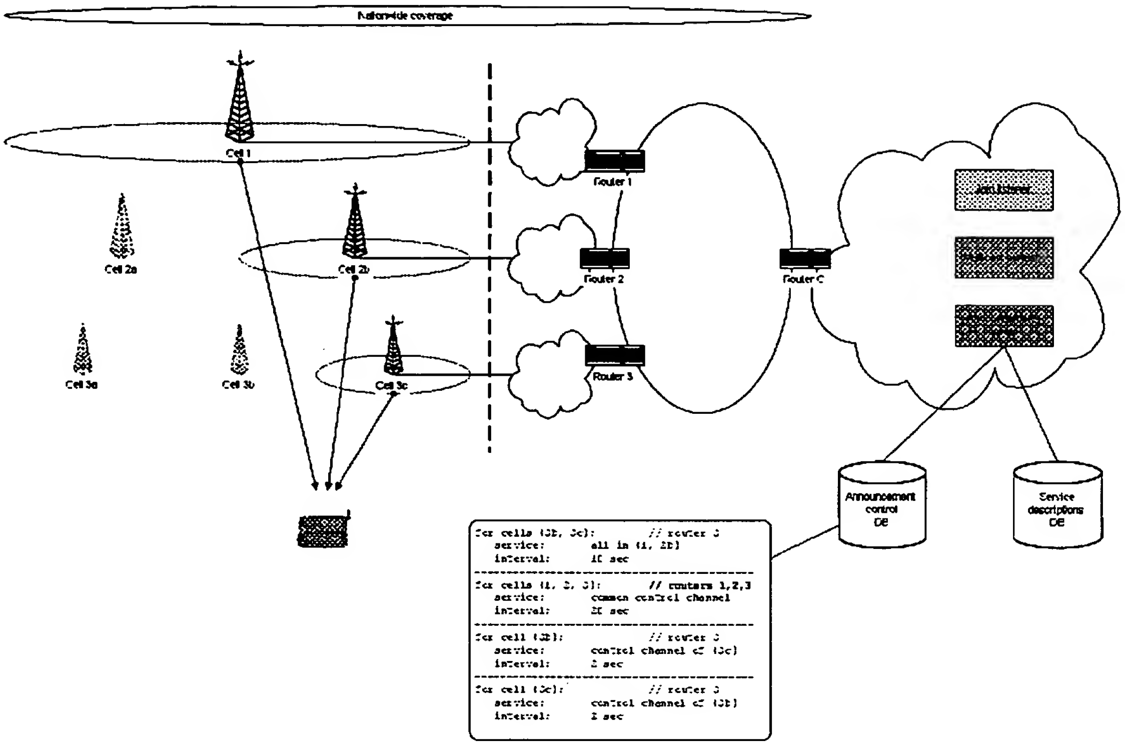
[1] Session Description Protocol, RFC-2327

[2] ASAB Requirements specification, ASAB-D1

[3] ASAB Protocol specification, ASAB-D2

PART 1 – SERVER FUNCTIONAL SPECIFICATION

3. CONTEXT



```
for cells (3b, 3c):           // router 3
   service:        all in (1, 2b)
   interval:       10 sec
-----------------------------------------
for cells (1, 2, 3):          // routers 1,2,3
   service:        common control channel
   interval:       20 sec
-----------------------------------------
for cell (3b):                // router 3
   service:        control channel of (3c)
   interval:       2 sec
-----------------------------------------
for cell (3c):                // router 3
   service:        control channel of (3b)
   interval:       2 sec
```

Figure 3–1, The reference network topology

The Figure 3–1 illustrates the reference network topology we consider in project ASAB. Additionally, the figure shows the logical placement of service annoucement server with respect to other entities in the network.

4. FUNCTIONAL ARCHITECTURE

[Toni]

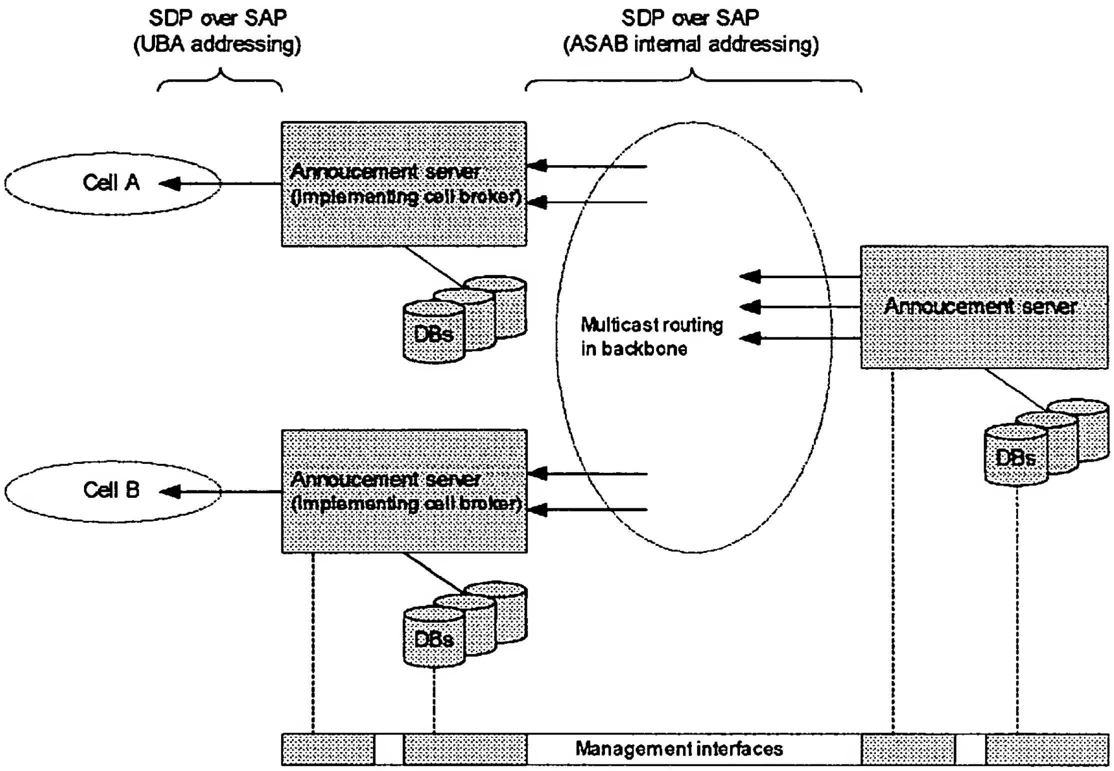


Figure 4–1, Functional architecture of the service announcing system

The functional architecture of the entire service annoucement system with is depicted in the Figure 4–1. The main functional components are:

Announcement server

The announcement server is the main component of the system. It takes care about composing the service announcement, sending them out as SDP over SAP and rescheduling them. The announcement server is associated with a set of databases, which define its configuration and behaviour.

Cell broker

There is one cell broker per one logical cell in the system. The cell broker is actually a system function, which can is implemented by the last hop announcement server associated with a particular cell. Depending on the configuuration, the announcement server can act as a plain annoucement server, a cell broker, or both.

The cell broker is in charge of what is announced in the cell it has control over. The cell broker receives SDP announcements from two sources. First, it receives announcements from the other announcement servers in the backbone as ASAB internal multicast transmissions. Second, the annoucement server is associated with a set of databases that desribe which service annoucements to compose and send.

Thus, the cell broker function is to decide which announcements should be transmitted to the actual cell. Also, the cell broker does translation between ASAB internal SAP multicast addresses to the multicast addressing used in UBA. Last, the cell broker ensures that the pace of announcement sending complies with the agreed bandwidth limits. This means that the cell broker acts as logical a filter-mapper-scheduler.

Database servers

The database servers contain four logical databases. First, the service description database contains the incredients for the basic SDP [1] service announcements. The second database, the announcement control information database contains the control information about how to announce services described in the first database. Third, the mapping databse contains the special cell-to-IP and connectivity mappings defined in ASAB protocol specification [3]. Last, the configuration database contains information of how the server is configured to work.

It is envisioned that every announcement server (whether implementing cell broker functionality or not) has a local database server associated. However, this is not required. An annoucement server could use a local database for configuration and connectivity mappings, for example. The same server could then use a centralised, remote database for fetching the service descriptions.

Management interfaces

Each of the main components has a management interface. The purpose of these interfaces is to allow system manager to monitor and adminstrate the server. Also, these interfaces provide other UBA system modules a way to inter-operate with the announcement server system.

PART 2 – SERVER TECHNICAL SPECIFICATION

5. ANNOUNCEMENT SERVER

[Toni]

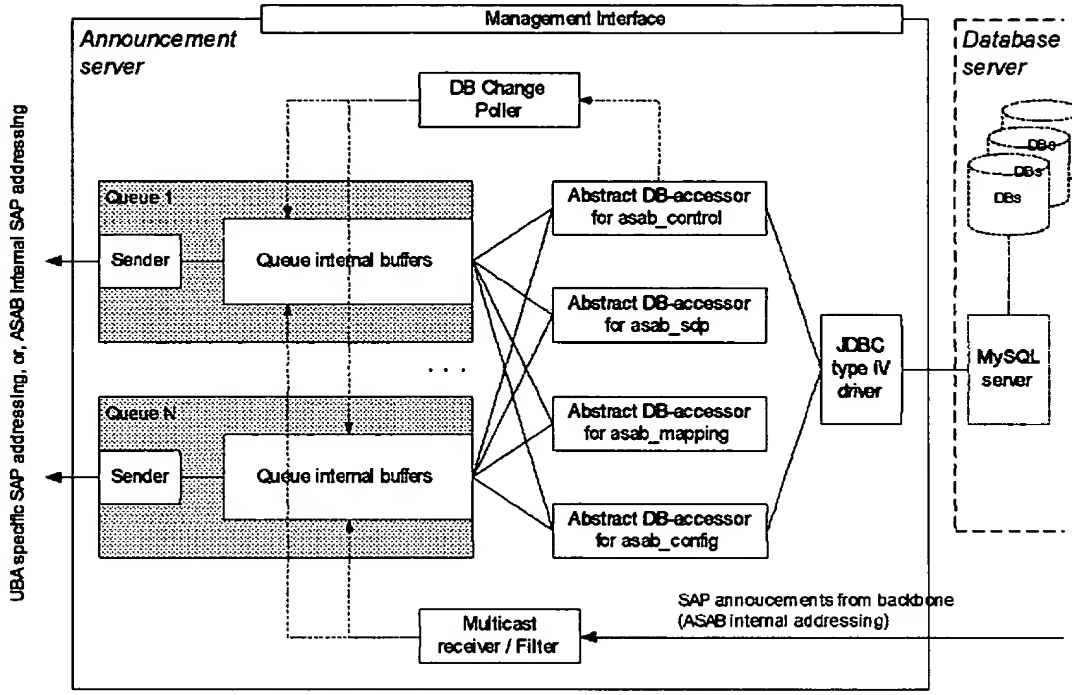

Management Interface
Announcement server
Database server
DB Change Poller
DBs
DBs
DBs
Queue 1
Sender
Queue internal buffers
. . .
Queue N
Sender
Queue internal buffers
Abstract DB-accessor for asab_control
Abstract DB-accessor for asab_sdp
Abstract DB-accessor for asab_mapping
Abstract DB-accessor for asab_config
JDBC type IV driver
MySQL server
UBA specific SAP addressing, or, ASAB internal SAP addressing
SAP annoucements from backbone (ASAB internal addressing)
Multicast receiver / Filter

Figure 5–1, Announcement server components

Components of the announcement server include

- Abstract database accessors
  - For control info
  - For basic sdp descriptions
  - For ASAB specific mapping information (extended SDP)
  - For server configuration info
- JDBC database driver
  - JDBC Type-IV, 100% Java
- DB Change poller
  - Polls the changes that happen in control DB

    - When a change occurs, the target queue is modified accordingly. The modification might be adding/removing an SDP or adding/removing a control element

- Queues
    - There are multiple queues in an annoucement server. The meaning of a queue is to enable higher level of grouping the announcements that are prepared at the server. The idea is that a queue represents somehow grouped announcements. In addition, SAP announcement that originate the same queue will have the same IP multicast address as destination.This way, annoucement servers that implement cell brokers can subscribe to get certain types of announcements by just joining the respective multicast groups.
    - The scheduling policy of the queue depend on a single parameter, bandwidth limit (bw_limit). If the limit is set to 0, the queue follows blindly the configuration given an the control_into DB. Otherwise the sender associated with the queue will requlate the pace of transmissions.
- Multicast receiver/filter, which contains
    - Multicast receiver and multicast-group-to-queue-mapper
    - This does essentially the same as DB change poller. This component installs new descriptions into correct queues as soon they are received.

Interfaces to other UBA system components and management

- Management interface

5.1 JDBC Drivers

[Jani]

5.2 Database accessors

[Jani]

5.3 Queue

[Toni]

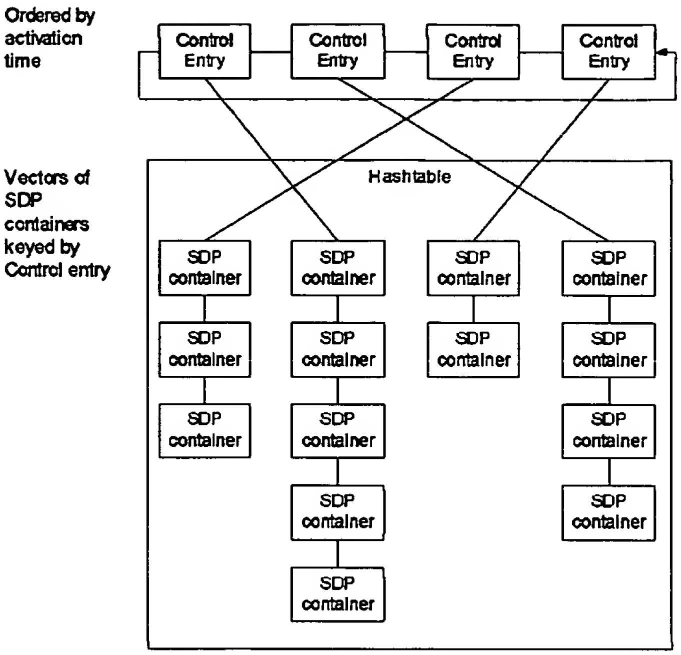

Ordered by activation time
Control Entry
Control Entry
Control Entry
Control Entry
Vectors of SDP containers keyed by Control entry
Hashtable
SDP container

Figure 5–2, Queue internal buffers

5.4 MulticastReceiverMapper

[Toni]

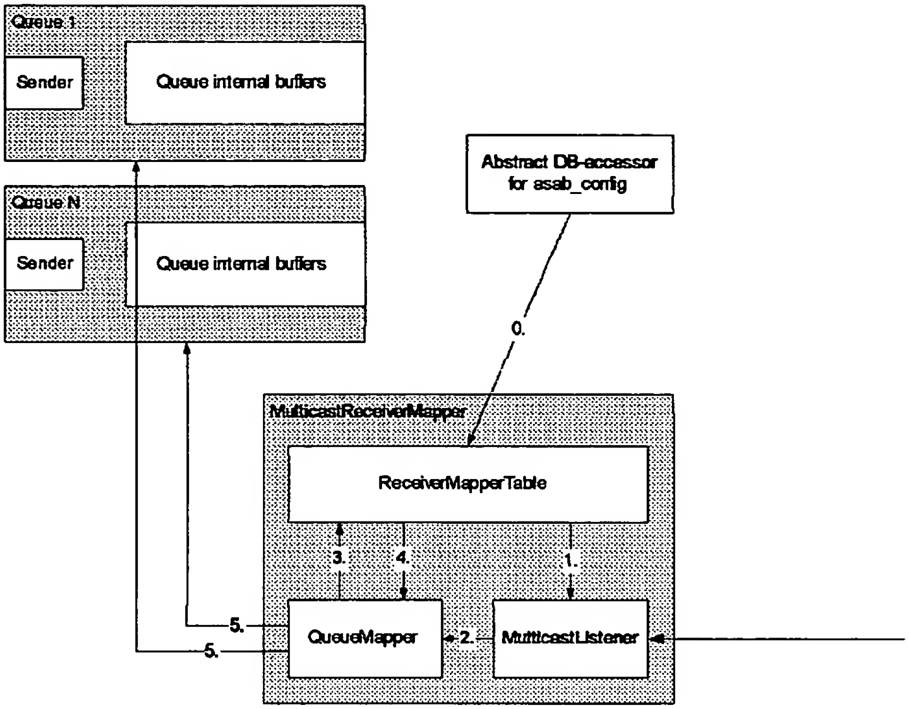


Figure 5–3, MulticastReceiverMapper operation

The MulticastReceiverMapper structure is depicted in the Figure 5–3. There are three main components in a MulticastReceiverMapper.

- **MulticastListener** listens for multicast SAP annoucements via a socket. The multicast groups are defined in server configuration database asab_config. The groups that the MulticastListener listens is sum of all the to-be-listened addresses in the database.

- **QueueMapper** maps the incoming multicast SAP packets to correct queues. It uses the ReceiverMapperTable to achieve that.

- **ReceiverMapperTable** contains mappings from a multicast address to a set of queues. In fact, the ReceiverMapperTable is the memory of the MulticastReceiverMapper.

Multicast address (ASAB internal SAP address)	Address range	Vector of references to target queues
224.10.10.2	1	{Queue1, Queue2, Queue4}
224.10.15.6	1	{Queue1, Queue3}
224.100.4.0	20	{Queue2, Queue4}

Table 5-1, Example of ReceiverMapperTable contents

The basic operation of MulticastReceiverMapper is annotated with numbered steps in the Figure 5–3.

0. Configuration information is inserted to ReceiverMapperTable (an example is shown in Table 5-1). The information originates from the database asab_config.
1. MulticastListener is initialised/updated with a set of to-be-listened multicast addresses.
2. MulticastListener sends a received multicast SAP announcement to QueueMapper for further mapping to queues.
3. QueueMapper queries the ReceiverMapperTable to find out which queues to send a copy of announcement.
4. ReceiverMapperTable returns a vector of references to target queues.
5. QueueMapperTable inserts a copy of the received packet to each queue referenced by the result vector in 4.

5.5 DbChangePoller

[Jani]

5.6 Cell broker function

The cell broker functionality is distributed in the announcement server to several places.

Queue configuration database holds ASAB-SAP to UBA-SAP mapping information

- What is the target multicast group of this queue
- Which multicast groups from the backbone will map to this queue

It also holds the filtering information

- If a ASAB internal multicast group does not appear in the queue configuration, the queue won't receive or relay any packets from that group

Sender-scheduler

- Normal announcement server queues handle this task.

5.7 File: Startup.properties

[TDB, Jani]

6. DATABASE SERVER

[Jani]

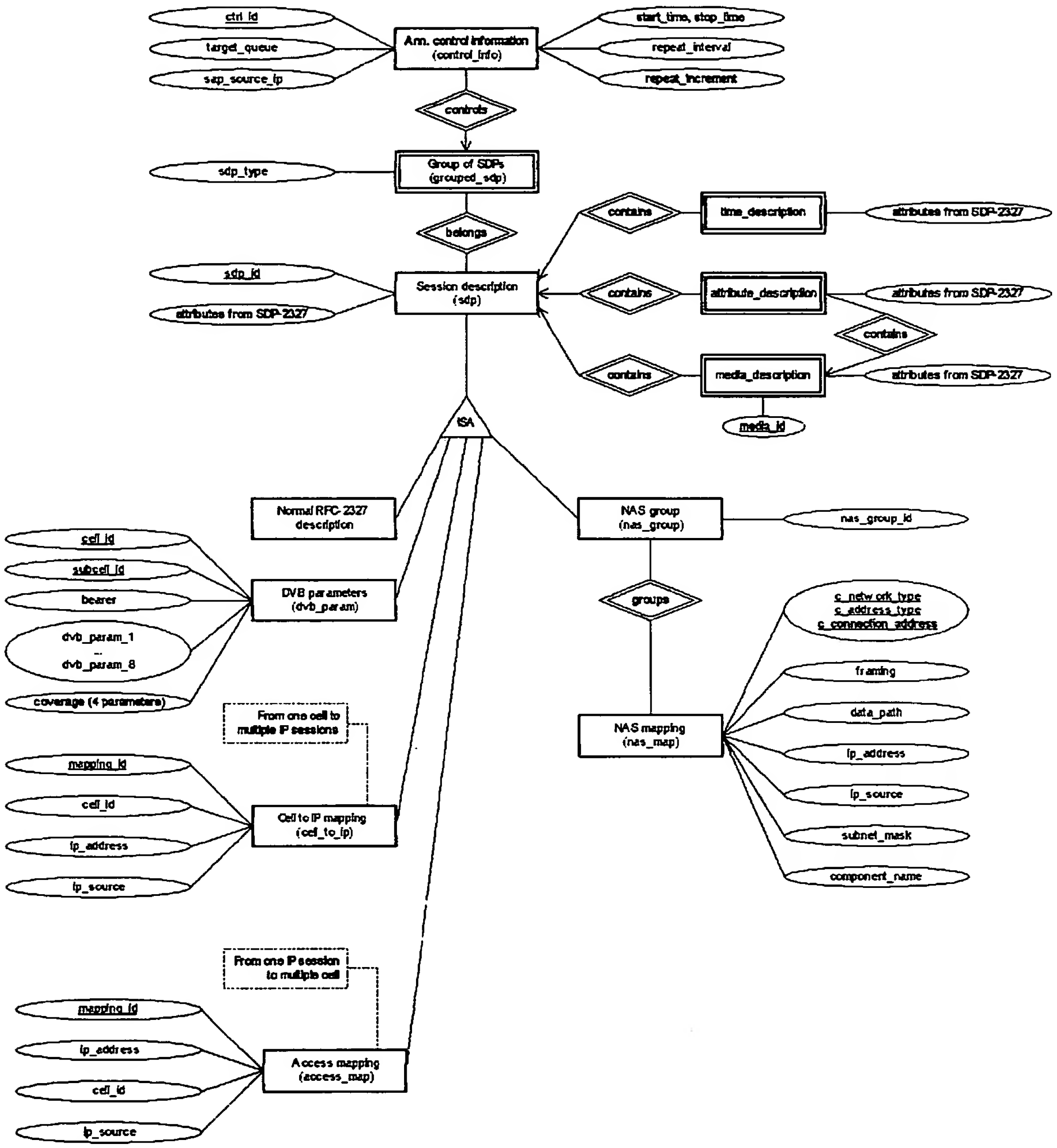


Figure 6–1, E/R-model of the information in the databases

The database server is simple. It only consists of a MySQL server and the databases. The E/R-model of the information in the databases is shown in the Figure 6–1.

This information is consequently divided into tables. Furthermore, there are four separate databases which hold the tables. The descriptrions per database are:

Database name	Description	Tables included
asab_control	Contains tables that define how the the announcements are grouped and scheduled.	control_info grouped_sdp multicast_listen
asab_sdp	Contains tables that define the content of service/session descriptions	sdp time_description attribute_description media_description
asab_mapping	Contains tables that define special, ASAB-specific mappings, such as cell-to-IP –mapping and NAS announcements	dvb_param cell_to_ip access_map nas_group nas_map
asab_config	Contains tables for server configuration	queue_config

The management interface to the databases is provided directly with the MySQL client.

6.1 Database tables for basic session descriptions

6.1.1 Service description directory (TABLE asab_sdp.sdp)

Field name	Key	Type	Notes
sdp_id	x	integer	unique key, not null
v_version		char	v, (protocol version)
o_user_name	x[2]	varchar	o, "-" | (user login)
o_session_id	x[1]	bigint	o, NTP
o_version		bigint	o, increase o_sessionId per modification
o_network_type	x[1]	varchar	o, "IN"
o_address_type	x[1]	varchar	o, "IN4"
o_address	x[1]	varchar	o, (address of the machine from which the session was created) | (FQDN)
s_session_name		varchar	s, "MP3 stream"
i_information		varchar	i*, "Nice music..."
u_uri		varchar	u*
e_email		varchar	e*, "foo@bar.com"
p_phone		varchar	p*, See RFC-2327
c_network_type		varchar	c*, "IN" (connection information – not required if included in all

[2] As defined in RFC-2327 (Session Description Protocol)

			media)
c_address_type		varchar	c*, "IP4" (connection information - not required if included in all media)
c_connection_address		varchar	c*, "224.2.17.12/127" (connection information - not required if included in all media)
b_bandwidth		varchar	b*, See RFC-2327
z_adjustment		varchar	z*, "2882844526 -1h 2898848070 0" **note:** ASAB groups all z subfields into one field
k_method		varchar	k*, "dear"
k_encryption_key		varchar	k*, See RFC-2327

6.1.2 Time description lookup table (TABLE asab_sdp.time_description)

Field name	Key	Type	Notes
sdp_id	x	integer	to which service description this belongs, not null
t_start_time		bigint	t, 2873397496
t_stop_time		bigint	t, 2873404696
r_repeat_interval		bigint	r*, 604800
r_active_duration		bigint	r*, 3600
r_offsets		varchar	r*, "0 90000" (zero or more repeat times)

6.1.3 Session attribute description lookup table (TABLE asab_sdp.session_attribute_description)

Field name	Key	Type	Notes
sdp_id	x	integer	to which service description this attribute belongs
media_id	x	integer	to which media description in session 'sdp_id' this attribute belongs (0 means session-level attribute)
a_name		varchar	a*, "connect"
a_value		varchar	a*, "1.2.3.4"

6.1.4 Media description lookup table (TABLE asab_sdp.media_description)

Field name	Key	Type	Notes
sdp_id	x	integer	to which service description this belongs
media_id	x	integer	part of key
m_media		varchar	m, "video"
m_port		integer	m, 49170
m_number_of_ports		integer	m, 2
m_transport		varchar	m, "RTP/AVP"
m_fmt_list		varchar	m, "31"
i_media_title		varchar	i, "Foobar"
c_network_type			c*, "IN" (connection information for media)
c_address_type		varchar	c*, "IP4" (connection information for media)
c_connection_address		varchar	c*, "224.2.17.12/127" (connection information for media)
b_bandwidth		varchar	b*, See RFC-2327
k_method		varchar	k*, "dear"
k_encryption_key		varchar	k*, See RFC-2327

6.2 Database tables for Service announcement control information

6.2.1 Service announcement directory (TABLE asab_control.control_info)

Field name	Key	Type	Notes
server_ip	x	varchar(15)	target server ip to which the control element refers
server_port	x	integer	target server port to which the control element refers
ctrl_id	x	integer	no duplicate control entries per server
target_queue		integer	to which queue to put the SDP packets that belong to the scope this announcement control information
sap_source_ip		varchar	IPv4 source address to appear in SAP header See SAP specification, chapter 3
start_time		bigint	Start time; when this control information activates, i.e. when to start announcing (in NTP)
stop_time		bigint	Stop time; when this control information de-activates, i.e. when to stop announcing (in NTP)
repeat_interval		bigint	Announcing interval when the announcements are active, i.e. between start_time and stop_time
repeat_increment		Bigint	0 = When stop_time reached -> delete this control entry <other> = When stop_time reached do the following: start_time := stop_time + repeat_increment

6.2.2 Grouped SDP table (TABLE asab_control.grouped_sdp)

Field name	Key	Type	Notes
server_ip	x	integer	target server ip to which the control element refers
server_port	x	integer	target server port to which the control element refers
ctrl_id	x	integer	no duplicate control entries per server
sdp_id		integer	Reference to asab_sdp.sdp (null only in multicast listen case)
ref_in_dvb_param		integer	Reference to asab_mapping.dvb_param (null if not used)
ref_in_cell_to_ip		integer	Reference to asab_mapping.cell_to_ip (null if not used)
ref_in_ip_to_cell		integer	Reference to asab_mapping.ip_to_cell (null if not used)
ref_in_multicast_listen		integer	Reference to asab_control.multicast_listen (null if not used)

6.2.3 ASAB internal SAP announcements from core network (TABLE asab_control.multicast_listen)

Field name	Key	Type	Notes
ref_in_multicast_listen		integer	
asab_sap_address		varchar	the ASAB internal multicast group to listen

6.3 Database tables for special mappings

6.3.1 DVB parameters (TABLE asab_mapping.dvb_param)

Needs asab_mapping.subcell

Field name	**Key**	**Type**	**Notes**
ref_in_dvb_param	x	integer	
c_address_type		varchar	c*, "IN4" or "SI"
cell_id	x	varchar(8)	Examples: "DVB/CELL SI **34567**", or, "DVB/CELL IN4 **12.13.14.15**"
bearer		varchar(8)	"a=bearer:**dvb-t**"
framing		varchar(8)	" a=framing:**dvb/mpe**"
dvb_param_1		varchar(8)	"a=dvb-t-bandwidth:**8**"
dvb_param_2		varchar(8)	"a=dvb-t-fft:**8**"
dvb_param_3		varchar(8)	"a=dvb-t-constellation:**16QAM**"
dvb_param_4		varchar(8)	"a=dvb-t-coderate:**2/3**"
dvb_param_5		varchar(8)	"a=dvb-t-guard-interval:**1/8**"
dvb_param_6		varchar(8)	"a=dvb-t-hierarchy:**none**"
dvb_param_7		varchar(8)	"a=dvb-t-hierarchical-priority:**high**"

6.3.2 DVB Subcell Information (TABLE asab_mapping.subcell)

Needs asab_mapping.dvb_param

Field name	**Key**	**Type**	**Notes**
cell_id	x	varchar(8)	Examples: "DVB/CELL SI **34567**", or, "DVB/CELL IN4 **12.13.14.15**"
m_media		varchar(8)	"m=**nas/none**"
subcell_id	x	integer	"a=subcell:**1** 450.2/60.3N/12.44E/3.1/2.5"
frequency		varchar(8)	"a=subcell:1 **450.2**/60.3N/12.44E/3.1/2.5"
coverage_param_1		varchar(8)	"a=subcell:1 450.2/**60.3N**/12.44E/3.1/2.5"
coverage_param_2		varchar(8)	"a=subcell:1 450.2/60.3N/**12.44E**/3.1/2.5"
coverage_param_3		varchar(8)	"a=subcell:1 450.2/60.3N/12.44E/**3.1**/2.5"
coverage_param_4		varchar(8)	"a=subcell:1 450.2/60.3N/12.44E/3.1/**2.5**"

6.3.3 Cell to IP mapping (TABLE asab_mapping.cell_to_ip)

Needs asab_mapping.xmap

Field name	Key	Type	Notes
ref_in_cell_to_ip	x	integer	
session_group	x	integer	reference to asab_mapping.xyz

6.3.4 Session to cell mapping (TABLE asab_mapping.ip_to_cell)

Needs asab_mapping.xmap

Field name	Key	Type	Notes
ref_in_ip_to_cell	x	integer	
cell_group	x	integer	ref to asab_mapping.xyz

6.3.5 xyz mapping (TABLE asab_mapping.xmap)

Used by asab_mapping.cell_to_ip

asab_mapping.ip_to_cell

Field name	Key	Type	Notes
cell_connection_addr_type		varchar	"IP4" | "SI"
cell_connection_address		varchar	"1.2.3.4" | "32765"
o_user_name		varchar	o, "-" | (user login)
o_session_id		bigint	o, NTP
o_network_type		varchar	o, "IN"
o_address_type		varchar	o, "IN4"
o_address		varchar	o, (address of the machine from which the session was created) | (FQDN)
session_group		integer	
cell_group		integer	

6.3.6 Group of NAS announcements (TABLE asab_mapping.nas_group) [To be finished...]

Field name	**Key**	**Type**	**Notes**
sdp_id	x	int	
nas_id		int	refers to asab_mapping.nas_map

6.3.7 NAS mapping (TABLE asab_mappings.nas_map) [To be finished...]

Field name	**Key**	**Type**	**Notes**
nas_id	x	int	
c_network_type	x		
c_address_type	x		
c_connection_address	x		
Framing		varchar	
data_path		varchar	
ip_address		varchar	
ip_source		varchar	
subnet_mask		int	
component_name		varchar	

6.4 Database tables for server configuration

6.4.1 Queue configuration for basic parameters (TABLE asab_config.queue_config)

Field name	Key	Type	Notes
server_ip	x	varchar(15)	target server ip to which the control element refers
server_port	x	integer	target server port to which the control element refers
queue_id	x	integer	identifier of the queue; not null
target_sap_ip		varchar	Destination IP address in packet carrying SAP messages from this queue
target_sap_port		integer	Target SAP port for the socket (this is for the future, if the queue needs to listen the annoucements itself)
bw_limit		integer	the bw limit (refer to SAP specification) in bits per second. If this is 0, there is no limit

7. MANAGEMENT INTERFACE

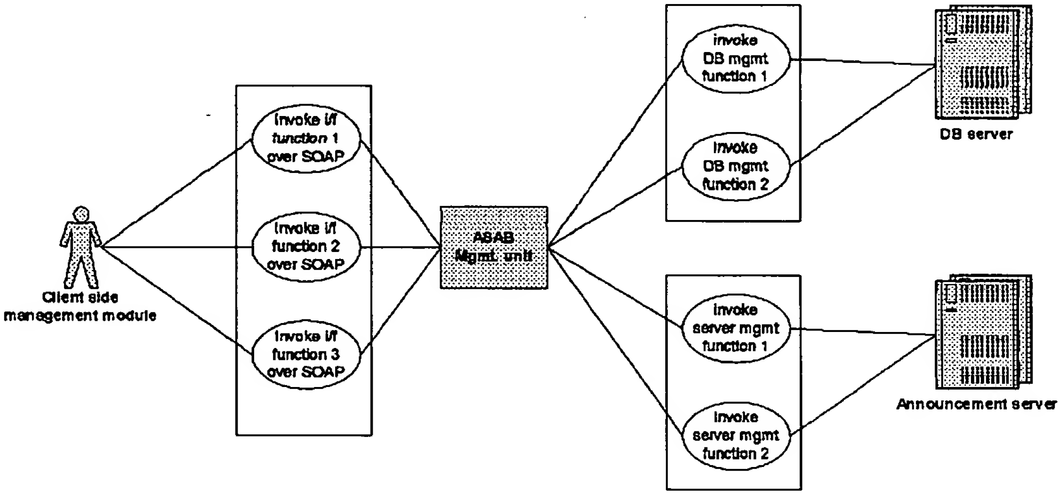


Figure 7–1, Overview of service&management interface operation

7.1 Management architecture

The management architecture bases on the the following principles:

- Through one management unit you can access and configure arbitrary many announcement servers and databases.
- Every component of the ASAB system can be managed as an independent unit. This allows e.g. every database to be configured independently without a running server (with a few exceptions, of course).

- No components are positional. In other words the management units, announcement servers and databases can be located in different servers.
- Every component is functional as stand-alone unit and thus independent of the management unit.
- The management unit is runtime configurable in certain limits.
- Every announcement server has only one asab_config, asab_control, asab_sdp and asab_mapping database. Still one database can be used by multiple announcement servers.

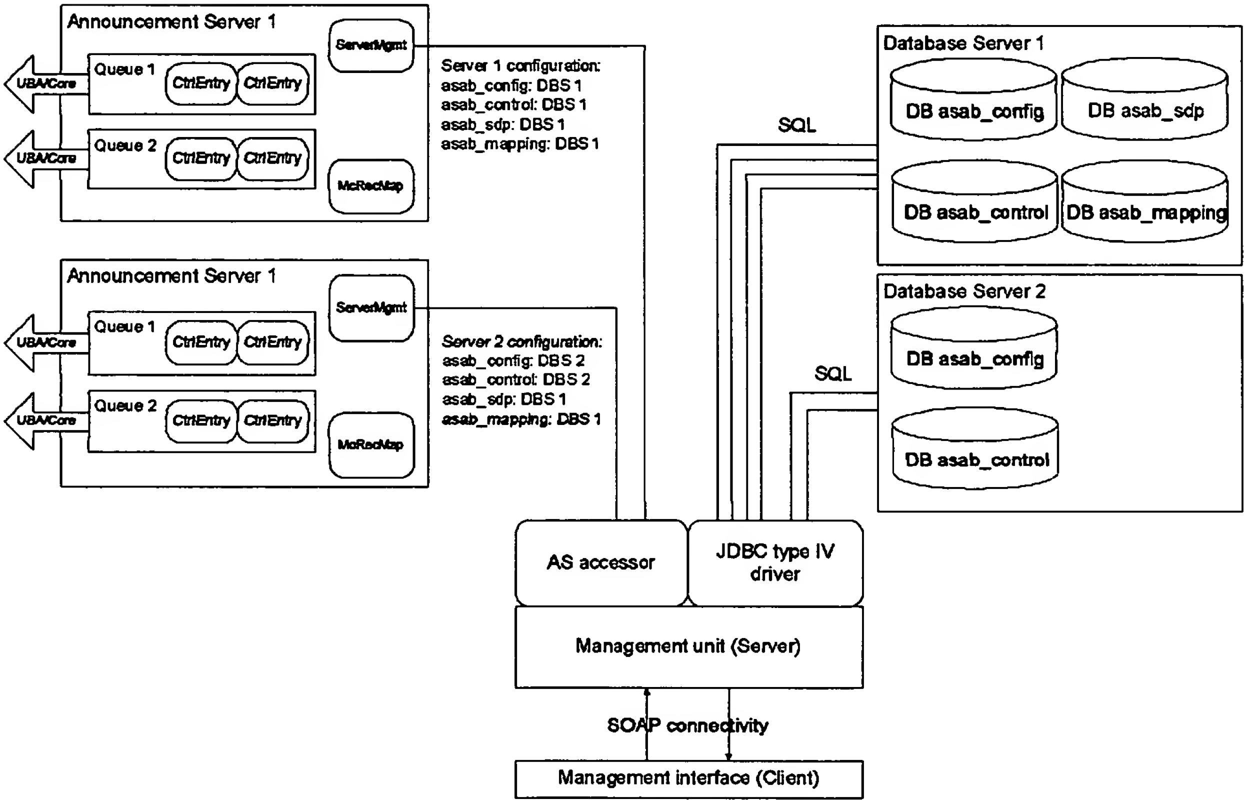


Figure 7–2, An example of a ASAB configuration

The management unit provides a remote management interface that you can control the ASAB system with. The data transfer between management unit and management interface is enabled by a point-to-point connecetion using SOAP RPC protocol.

Connections between management unit and databases are simple SQL connections through Java JDBC drivers.

AS and management unit connection protocol is still open, but SOAP seems to be strongly suggested also here.

The server's independency of the management interface is enabled by assingning a boot sequence which first searches the configuration locally, second inquires it from management unit and is still no confdiguration is found, starts as null.

7.2 Management API

The complete class diagram of the ASAB management inteface. Note that there's only six classes with services while other classes are more like data containers.

asab_mgmt_diag_1_1.ps

7.2.1 <AsabManagementInterface>

The management interface for ASAB servers & databases. One instance of this class can be used to manage arbitrary many ASAB servers. Each announcement server needs four databases (asab_config, asab_control, asab_sdp, asab_mapping) to be able to sent announcements. Despite databases and announcement servers can be accessed if all four databases are not configured. **ServerIdentifier addServer(ServerConfiguration)** Add a new server under the management. Use modifyServerConfiguration to modify a server configuration. Here and in the following methods in this class the server means the complex of AS and the databases associated with it. It's mandatory that the servers IP address & port is defined in ServerConfiguration, although the defined server doesn't need to yet exist. Database definition in the configuration are optional. Naturally the accessibility of databases depends on this.

void removeServer(ServerIdentifier)
Remove a server under the management. Only the configuration information of the server & databases is removed - the action of the server and the databases is not affected.

Vector getServerIdentifiers()
Get a vector of the ServerIdentifiers that are added to this management interface.

ServerConfiguration getServerConfiguration(ServerIdentifier)
Get the configuration of the server.

void modifyServerConfiguration(ServerIdentifier, ServerConfiguration)
Modify the configuration of a server.

ServerIdentifier readServerConfiguration(String, int)
Connect to an existing stand-alone announcement server and read its configurations. This method is similar to addServer, except the configurations are not given by the user of the interface but instead from a server. Arguments are IP address (String) and port (int).

ServerAccessor getServerAccessor(ServerIdentifier)
Get the accessor for a server.

ConfigDbAccessor getConfigDbAccessor(ServerIdentifier)
Get the accessor for a database asab_config of a specific server.

ControlDbAccessor getControlDbAccessor(ServerIdentifier)
Get the accessor for a database asab_control of a specific server.

SdpDbAccessor getSdpDbAccessor(ServerIdentifier)
Get the accessor for a database asab_sdp of a specific server.

MappingDbAccessor getMappingsDbAccessor(ServerIdentifier)
Get the accessor for a database asab_mapping of a specific server.

ServerConfiguration
-String server_ip
-int server_port
-DbConfiguration configDbConfig
-DbConfiguration controlDbConfig
-DbConfiguration sdpDbConfig
-DbConfiguration mappingDbConfig

DbConfiguration
-String dbDriver
-String dbMgmtSystem
-String dbHostIp
-int dbPort
-String dbName
-String username
-String passWord

7.2.2 <ServerAccessor>

This accessor accesses the announcement server.

ServerState getServerState()
Get the current state of the server.

void startQueue(QueueIdentifier)
Start a queue.

void stopQueue(QueueIdentifier)
Stop a queue.

void disableMcReceiverMapper()
Stop listening the multicast addresses configured.

void enableMcReceiverMapper()
Start listening the multicast addresses configured. The sap packet received are sent forward to queues configured.

ServerState
-QueueIdentifier[] runningQueues
-Boolean isMcReceiverMapperRunning
-...

7.2.3 <DbAccessor>

In addition to their own methods, all four database accessors ConfigDbAccessor, ControlDbAccessor, SdpDbAccessor, MappingDbAccessor implement similar database functions, which are listed here.

void appendDb(DbAccessor)
Appends another database to this database. Appending a database to an empty database is in practice duplicating. This operation doesn't affect the database which accessor is given as an argument.

void emptyDb()
Empty the database.

Identifier add(Content)
Add a Content to the database. When modifying an existing Content, use modifyContent instead. As return you get an Identifier referring to the Content

void remove(Identifier)
Remove a Content from the database.

Vector getIdentifiers()
Get a vector of identiers of the Contents in the database.

Content getContent(Identifier)
Get a Content element that the Identifier refers to.

void modifyContent(Identifier, Content)
Replace the Content (in database) that Identifier refers to with Content given as argument.

Here's how the terms above differ in different accessors:

DbAccessor is
in ConfigDbAccessor: ConfigDbAccessor
in ControlDbAccessor: ControlDbAccessor
in SdpDbAccessor: SdpDbAccessor
in MappingDbAccessor: MappingDbAccessor

Identifier is
in ConfigDbAccessor: QueueIdentifier
in ControlDbAccessor: ControlEntryIdentifier
in SdpDbAccessor: SdpPacketIdentifier
in MappingDbAccessor: 1) DvbParamIdentifier, 2) CellToIpMappingtIdentifier, 2) CellIdentifier

Content is
in ConfigDbAccessor: QueueConfiguration
in ControlDbAccessor: ControlEntry
in SdpDbAccessor: SdpPacket
in MappingDbAccessor: 1) DvbParam, 2) CellToIpMapping, 3) Cell

7.2.4 <ConfigDbAccessor>

This DbAccessor accesses the database asab_config.

QueueConfiguration
-String target_sap_ip
-int target_sap_port
-int bw_limit

7.2.5 <ControlDbAccessor>

This DbAccessor accesses the database asab_control.

void insertCtrlEntryIntoQueue(CtrlEntryIdentifier, QueueIdentifier)
Set a control entry to a queue. The queue and control entry must have been added to the databases of the same announcement server before an control entry can be addressed to a queue.

void removeCtrlEntryFromQueue(CtrlEntryIdentifier, QueueIdentifier)
Remove a control entry from a queue.

Vector getCtrlEntries(QueueIdentifier)
Get control entries inserted into a queue. Vector contains CtrlEntryidenftifier objects.

QueueIdentifier getTargetQueue(CtrlEntryIdentifier)
Get queue identifier by control entry.

void insertToControlEntry(Object, CtrlEntryIdentifier)
Attach an object to acontrol entry. The object given as can be either SdpPacketIdentifier, McRecMapConfig, CellToIpIdentifier or DvbParamMApping.

void removeFromControlEntry(Object, CtrlEntryIdentifier)
Remove an object from a control entry. The object given as can be either SdpPacketIdentifier, McRecMapConfig, CellToIpIdentifier or DvbParamMapping.

Vector getSdpPacketIdentifiers(CtrlEntryIdentifier)
Returns a vector of SdpPacketIdentifier objects.

Vector getMcReceiverMapperConfigs(CtrlEntryIdentifier)
Returns a vector of McRecMapConfig objects.

Vector getDvbParamMappings (CtrlEntryIdentifier)
Returns a vector of DvbParamMapping objects.

Vector getCellToIpMappings(CtrlEntryIdentifier);
Returns a vector of CellToIpIdentier objects.

Vector getMcReceiverMapperCtrlEntries()
Get a vector of CtrlEntryIdentifier objects that contain McReceiverMapperConfig objects.

Vector getDvbParamMappingCtrlEntries()
Get a vector of CtrlEntryIdentifier objects that contain DvbParamMapping objects.

Vector getCellToIpMappingCtrlEntries()
Get a vector of CtrlEntryIdentifier objects that contain CellToIpMappingIdentifier objects.

```
CtrlEntry
 -String sap_source_ip
 -long start_time
 -long stop_time
 -long repeat_interval
 -long repeat_increment
```

7.2.6 <SdpDbAccessor>

This DbAccessor accesses the database asab_sdp.

```
SdpPacket
 -int sdp_type
 -String vVersion
 -String oUsername
 -long oSessionId
 -long oVersion
 -String oAddressType
 -String oAddress
 -String sSessionName
 -String information
 -String uUri
 -StringeEmail
 -String pPhone
 -String cNetworkType
 -String cAddressType
 -String cConnectionAddress
 -String bBandwidth
/* Timings which belong into this sdp packet */
 -Vector tTimings
 -String zAdjustement
 -String kMethod
 -String kEncryptionKey
/* Session attributes which belong into this sdp packet */
 -Vector aSessionAttributes
/* Media descriptions which belong into this sdp packet */
 -Vector mMediaDescriptions;
```

MediaDescription
-int mediaId
-String mMedia
-int mPort
-int mNumberOfPorts
-String mTransport
-String mFmtList
-String iMediaTitle
-String cNetworkType
-String cNetworkAddress
-String cConnectionAddress
-String bBandwidth
-String kMethod
-String kEncryptionKey

TimeDescription
-long tStartTime
-long tStopTime
-long rRepeatInterval
-long rActiveDuration
-String rOffsets

SessionAttribute
-MediaDescription mediaId
-String aName
-String aValue

7.2.7 <MappingDbAccessor>

This DbAccessor accesses the database asab_mapping.
Note: The method hierarchy in this accessor is still under development.

DvbParamMapping addDvbParamMapping()
Generate a new DvbParamMapping object. The object is used to map DvbParams to CtrlEntrys.

void removeDvbParamMapping(DvbParamMapping)
Remove a DvbParamMapping.

Vector getDvbParamMappings()
Gets a vector of DvbParamMapping objects.

DvbParamMapping is ony an unique identifier given by the database. For the time being all mapping datastructures must be constructed manually.

DvbParam
-DvbParamMapping ref_in_dvb_param
-CellIdentifier cell_id
-String bearer
-String framing
-String dvb_param_1
-String dvb_param_2
-String dvb_param_3
-String dvb_param_4
-String dvb_param_5
-String dvb_param_6
-String dvb_param_7

SubCell
-int subcell_id
-String m_media
-String frequency
-String coverage_param_1
-String coverage_param_2

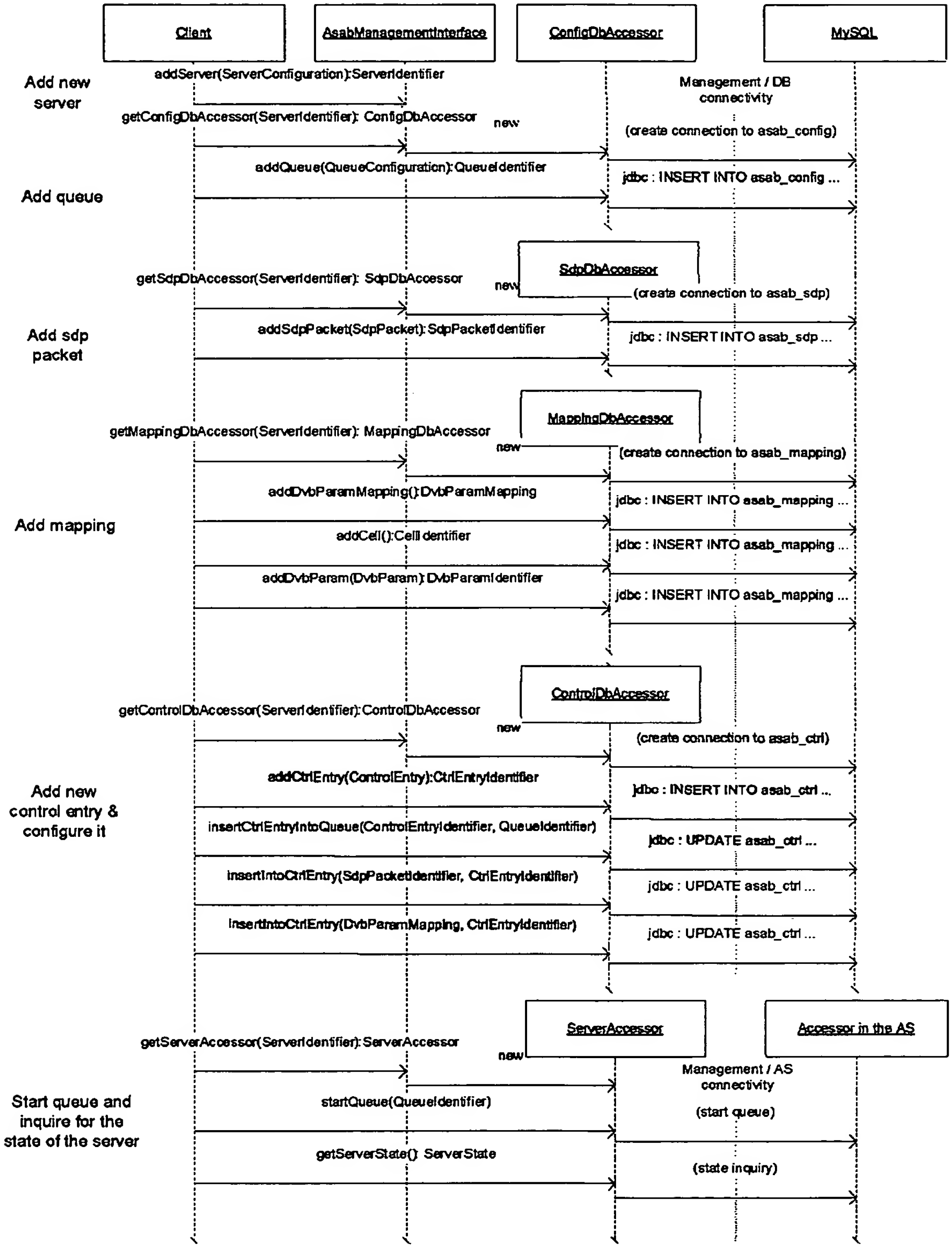
```
 -String coverage_param_3
 -String coverage_param_4

CellToIp
 -CellIdentifier cell_id
 -DvbParamIdentifier sdp_cell_id
 -String ip_address
 -String ip_source

Cell
 -String c_address_type
 -String cell_id;
/* A vector of SubCell objects. At least one subCell is needed. */
 -Vector subCells
```

7.3 Example case

Configuring announcement server from the beginning to send one session announcement with a DVB cell paramemeter announcement included


Client
AsabManagementInterface
ConfigDbAccessor
MySQL
Add new server
addServer(ServerConfiguration):ServerIdentifier
Management / DB connectivity
getConfigDbAccessor(ServerIdentifier): ConfigDbAccessor
new
(create connection to asab_config)
addQueue(QueueConfiguration):QueueIdentifier
jdbc : INSERT INTO asab_config ...
Add queue
SdpDbAccessor
getSdpDbAccessor(ServerIdentifier): SdpDbAccessor
new
(create connection to asab_sdp)
addSdpPacket(SdpPacket): SdpPacketIdentifier
jdbc : INSERT INTO asab_sdp ...
Add sdp packet
MappingDbAccessor
getMappingDbAccessor(ServerIdentifier): MappingDbAccessor
new
(create connection to asab_mapping)
addDvbParamMapping():DvbParamMapping
jdbc : INSERT INTO asab_mapping ...
Add mapping
addCell():CellIdentifier
jdbc : INSERT INTO asab_mapping ...
addDvbParam(DvbParam): DvbParamIdentifier
jdbc : INSERT INTO asab_mapping ...
ControlDbAccessor
getControlDbAccessor(ServerIdentifier):ControlDbAccessor
new
(create connection to asab_ctrl)
addCtrlEntry(ControlEntry):CtrlEntryIdentifier
jdbc : INSERT INTO asab_ctrl ...
Add new control entry & configure it
insertCtrlEntryIntoQueue(ControlEntryIdentifier, QueueIdentifier)
jdbc : UPDATE asab_ctrl ...
insertIntoCtrlEntry(SdpPacketIdentifier, CtrlEntryIdentifier)
jdbc : UPDATE asab_ctrl ...
insertIntoCtrlEntry(DvbParamMapping, CtrlEntryIdentifier)
jdbc : UPDATE asab_ctrl ...
ServerAccessor
Accessor in the AS
getServerAccessor(ServerIdentifier): ServerAccessor
new
Management / AS connectivity
Start queue and inquire for the state of the server
startQueue(QueueIdentifier)
(start queue)
getServerState(): ServerState
(state inquiry)

PART 3 – SERVER TEST SPECIFICATION

This part lists all the test cases for ASAB server. The tests include both component tests as well as functional tests.

8. COMPONENT TESTS

8.1 Test drivers

For component tests, there are test drivers for each tested module/component. The tests belong to package nrc.asab.tests. For example class nrc.asab.server.Queue has a test driver nrc.asab.tests.Queue_testdriver.

The driver classes are runnable. One should use test number(s) as parameters. For example, the following command runs tests 5-10 for class Queue.

```
java nrc.asab.tests.Queue_testdriver 5 10
```

See the Javadoc-documentation for test cases and further information.

8.2 Testing coverage

[TBD]

This is to be removed:

Test number	Test description
1	Test queue instantiation with several different parameter setups (correct, null, invalid)
2	Test queue modification: adding and removing valid and invalid control entries
3	Test queue modification: adding and removing valid and invalid SDPContainers

9. FUNCTIONALITY TESTS

THESE ALL REQUIRE A *SIMPLE* Script + TEST CLIENT

a. from scratch: install and run the server with test setup

- provide here name of the script and explain what it does (in 2 sentences)

b. regulated output speed

c. check multiple queues sending same announcements

d. check multiple queues sending different annoucements protocol tests:

tests that validate protocol:

ANNEX A - TERMS OF REFERENCE

This chapter defines the terms used in ASAB project. In the following definitions,.

Announcement control information	Annoucement control information controls how the service descriptions should be announced.
Announcement Server	Server in the distribution network or in the UBA core network, which creates service announcements and sends those to correct cells/links.
Service	The words service and session are used in a mixed way Service is something network offers and provides to end user. Services can be divided into individual services which are meant for one user or multiparty services which are aimed to two or more users. Examples of individual services are for example: video-on-demand and network access (aka. connectivity, or bit pipes with different characteristics). Multiparty services can be divided to private and public services. Examples of multiparty services are broadcast news service, broadcast file distribution service, multicast of web pages, mp3 distribution by broadcast, etc.
Service announcement	Also called session announcement due to the close relevance to SAP. The service announcement is a message that contains service/session description. The service announcement is the vehicle to convey the description from announcer to potential service users.
Service description	Also called session description due to the close relevance to SDP. The service description defines the type of service and other parameters related to it.

ANNEX B – UML CLASS DIAGRAM OF THE SERVER

To be added

Patent-Agency Banner-Witcoff (EXT-RES/Washington)

From: Paila Toni (Nokia-NVO/Helsinki)
Sent: Mon 7/30/2001 8:01 AM
To: Poikela Jani (Nokia-NRC/Helsinki)
Cc:
Subject:
Attachments: ASAB-D3-Server_046.doc(542KB)

<<ASAB-D3-Server_046.doc>>

NOKIA

CONFIDENTIAL

INVENTION REPORT

Title of invention: **A method for performing handover for multicast session in uni-directional access system.**	**INVENTION REPORT RECEIVED** Code: 19377 — Patent Committee: NVO/NRC
THE DESCRIPTION OF THE INVENTION MUST BE ATTACHED	Place: Helsinki — Date: 04.09.2001 Signature:

Inventor's name, employee number, title and nationality: *)	Home Address: *)	Business Unit and cost centre:
Toni Paila, 10000517, Research Engineer, Finnish	Everstinkuja 1 c 66 02600 Espoo Finland	NRC 1007950

Line Manager(s): Kari A. Rissanen

Project : *) ASAB	Project Manager: Toni Paila

Office address: *) NRC Ruoholahti A427

Phone: *) +358718037389	Fax: *) +358718036856

The invention becomes public on:

I am/ We are the sole/ and original inventor(s) of this invention.

The company may, by virtue of applicable legislation, be entitled to full or partial rights to the invention.
I/ We acknowledge my/ our obligation to sign as inventor(s) all documents that may be required for protecting the invention in different countries.

Applicable to inventions made by inventors employed in FI, DK, DE and SE only.
Unless the inventor requests the Invention Report to be responded to within four (4) months from the date this Invention Report is received or such other period as the mandatory provisions of the applicable local law may otherwise require, the inventor consents to the right of the employer to use a reasonable period of time for the evaluation of the invention. A reasonable period of time may exceed four (4) months.
☒ *I/ We request that the Invention Report be responded to within four (4) months.*

Date:
Signature(s) of Inventor(s):

*) See the instructions

I have read and understood the invention described in this Invention Report

Date:
Signature of Manager

INSTRUCTIONS FOR COMPLETING THE INVENTION REPORT

This Invention Report form is used in cases where an invention has been made by an employee of the Company. This Invention Report is confidential. Only the Patent Department may make copies of signed Invention Reports in order to request opinions or reply to the inventor(s).

The inventor completes the Invention Report and the description of the invention. The inventor does not fill in the 'Invention Report received' field. This field is filled in by the Patent Department. The Invention Report must have the names of all the inventors and their home addresses. If there is not enough space for all the names, addresses etc, please write them on a separate attachment. The first mentioned inventor is assumed to be the contact person in matters conceming the Invention Report. In the fields of office address, phone and fax, please fill in the contact person's information. Fill in the project field, if the invention is made in a project. The original Invention Report is signed by all inventors. Each page of the original Invention Report is signed by a Manager. In case it is difficult to obtain Manager's signature your Patent Department will take care of it.

It is suggested that the Invention Report and the description of the invention should be filled in as thoroughly as possible. If drawings or other kind of information cannot be attached to this form, they should be delivered separately.

The signed Invention Report is given directly to the local or business unit's Patent Department. Invention Report should also be sent by E-mail to the Patent Department. The Patent Engineer will inform the inventor of receiving the Invention Report. The Patent Engineer will obtain any expert opinions needed to properly evaluate the invention, will procure the Company's decision and inform the inventor accordingly.

DESCRIPTION OF THE INVENTION

1. Field and background of the invention

The field of the invention is data networking in unidirectional access systems. More presicely the invention relates to digital broadcast systems capable of broadcasting datagrams and receivers capable of receiving those, respectively. The invention will be explained in terms of TCP/IP networking suite, but a skilled person knowing the area can easily generalise the main ideas to suit any form of data networking.

The background of invention lies in two NRC projects, DRiVE and ASAB. In DRiVE we design and specify a so called hybrid radio network, which by definition consist of multiple, possibly administratively independent, standalone radio access systems. In DRiVE the main question is how to offer the end users multimedia services over heterogenous radio access systems transparently. Also, the goal is to make the service provision cost-efficient, both for end user and the network operators. The multimedia services here may consist of both multicast and unicast services. This invention, however, is applicable for multicast services.

In project ASAB we desing and implement session (also called service) announcement facility for a broadcast system. The work contains two main parts. First, there is a service announcement server. Second, we design extensions to Session Description Protocol to make it capable of expressing more than just basic IP connectivity in fixed networks. Such extended announcements describe, for example, physical parameters of a DVB-T cell (frequency, MAC, and other link-level parameters). In addition, the announcements descibe logical mappings that the user can use to find out how to reach the session he is interested. For example, given a multicast IP session, in which physical cells is it reachable. This mapping can also appear in reverse direction: given a physical cell, which multicast IP sessions are supported in that. This, the protocol specification work in ASAB is highly important for this invention.

2. A summary of the invention

The invention presents a way a (mobile) end user of a broadcast digital access system can perform cell-to-cell hand-over while preserving continuity of service. Here the service is assumed to be IP multicast session. It is easier to figure out the invention if one assumes that the multicast session is receive-only. However, the invention is equally applicable to multicast sessions that are not receive-only.

In the invention, there is a (mobile) end user that tunes to a digital broadcast data bearer. First user gets logical mapping messages that announce a presence of a multicast session. The user joins the session and starts receiving it. While receiving, the continuosly received logical mapping messages keep him updated about contents of the neighbouring (horizontal or vertical) cells. When reception of the current bearer signal goes down, has errors, or fades out, the user uses the gathered logical mappings to select a new physical or logical cell to attach to. After this the user joins the session and starts receiving it.

3. Describe the problem which the invention overcomes

The problem is best explained with an example. When the user is moving he will pass a sequence of broadcast cell coverage areas. The user is receiving a session in one cell that he has tuned. When the user goes beyond the edge of the coverage area the reception will fade out if nothing is done. Surely, there are cases the user would like to preserve the session continuity.

4. How was the problem solved earlier?

One way is to manually search for a new bearer.
DVB-T standards might have some simple mechanism for performing some kind of DVB-T specific handover. (For an answer: please consult DVB-T specialist)

5. How does the invention improve earlier solutions? Advantages and disadvantages of the invention?

Enables end user to make more intelligent selections based on (possibly extensive) learned knowledge base.

6. Drawings and brief description of the drawings

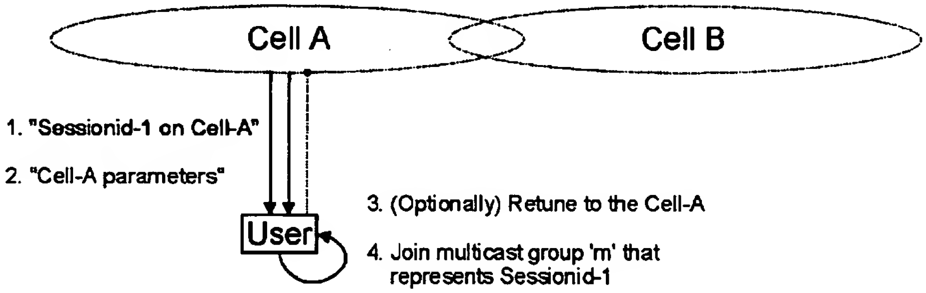

Cell A
Cell B
1. "Sessionid-1 on Cell-A"
2. "Cell-A parameters"
3. (Optionally) Retune to the Cell-A
User
4. Join multicast group 'm' that represents Sessionid-1

Figure 1

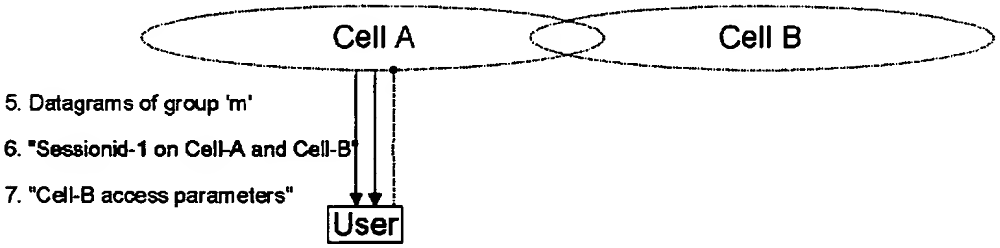

Cell A
Cell B
5. Datagrams of group 'm'
6. "Sessionid-1 on Cell-A and Cell-B"
7. "Cell-B access parameters"
User

Figure 2

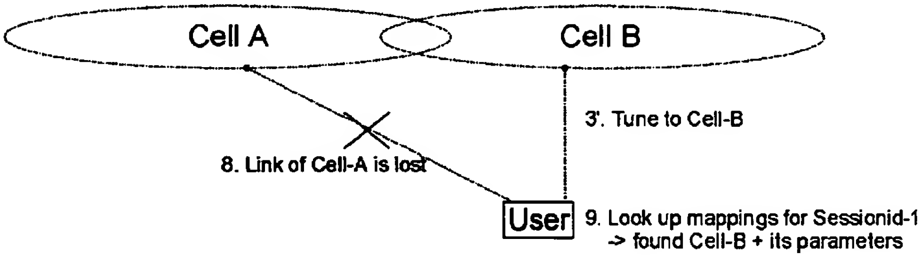

Cell A
Cell B
3'. Tune to Cell-B
8. Link of Cell-A is lost
User
9. Look up mappings for Sessionid-1
-> found Cell-B + its parameters

Figure 3

I have read and understood the invention described in this Invention Report

Date:
Signature of Manager

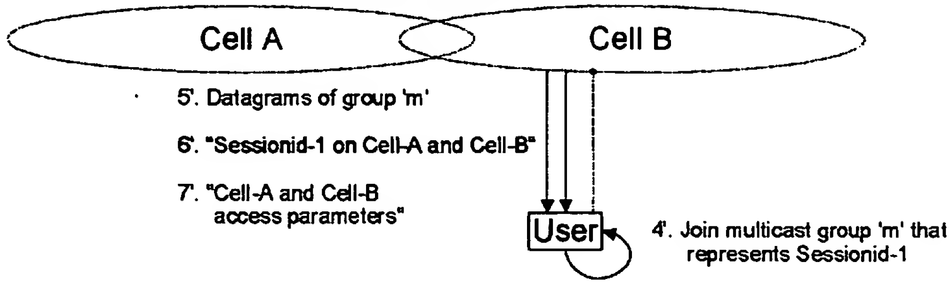

Cell A
Cell B
5'. Datagrams of group 'm'
6'. "Sessionid-1 on Cell-A and Cell-B"
7'. "Cell-A and Cell-B
access parameters"
User
4'. Join multicast group 'm' that
represents Sessionid-1

Figure 4

There are four figures (Figures 1 to 4) that present the sequence of events and actions that take place according to this invention. The following explanation details the sequence. Below, the 'user' is assumed to be intelligent service browser or equivalent – not the actual end user of the terminal equipment.

0. First, the user is attached to Cell-A and is receiving a logical announcemnt channel. This can be either predefined or dynamically configure IP multicast address. The broadcast network makes session announcements and mapping announcements available on this logical announcement channel.

1. User receives SDP announcement of session that has identifier Sessionid-1. There is also a mapping that tells that Sessionid-1 is available in Cell-A as well as in Cell-B.

2. User receives detailed link-level access parameters of Cell-A.

3. User optionally retunes to Cell-A (in case of DVB-T, user might need to change the MAC address or PID in the receiver end)

4. User joins the multicast group 'm' that was announced to represent Sessionid-1. Note that because user does not have an uplink, the join message is merely registered the operating system and the IP stack. However, it does not send any concrete IGMP join message anywhere.

5. User starts to receive datagrams of multicast group 'm' on Cell-A

6. While receiving group 'm', the user still receives session announcements and logical mappings. In this message, for example, Sessionid-1 is announced together with information that the Sessionid-1 is available on Cell-A as well as on Cell-B.

7. User receives detailed link-level access parameters of Cell-B

8. Reception of Cell-A signal may be lost for variuos reasons. The user may have left the coverage area, the Cell-A transmitter may experience a malfunction, there may be interference from some other source, etc.

9. User looks up the received mappings for Sessionid-1. He finds that Sessionid-1 is available on Cell-B. User also looks up for Cell-B and learns the detailed link-level access parameters.

3'. User tunes to Cell-B. (Note, from this point the logic follows numbering starting from 3.)

4'. User joins the multicast group 'm' that was announced to represent Sessionid-1.

5'. User starts (continnues) to receive datagrams of multicast group 'm' on Cell-B

6'. While receiving group 'm', the user still receives session announcements and logical mappings. In this message, for example, Sessionid-1 is announced together with information that the Sessionid-1 is available on Cell-A as well as on Cell-B.

7. A more detailed description of the invention (if known at the moment)

See 6 above.

8. Explain, how the invention is/can be implemented. Which would be the best mode of implementation?

If the announcement capability already exists the implementation will have impact to end users, only. There are two ways to implement. First, as operating system level function, or the as intelligent service browser (preferred).

9. Explain how we can recognise if a competitor is using the same product/feature?

-

10. Is it planned to use the invention in a Nokia product? If so, when and in which product? Is the invention standard related?

NVO/NEW/IPDC and project ASAB are designing and implementing announcement server that is capable of performing the announcing system. That system might become a part of Nokia product by the end of year 2002.

11. Abbreviations

ASAB	Advanced Service Announcement for Broadcasting
DVB-T	Digital Video Broadcast Terrestrial
SDP	Session Description Protocol

12. Any further comments

-

I have read and understood the invention described in this Invention Report

Date:
Signature of Manager

| ✉ | ✉ | ✉ | ✉ | ✉ | X | ▲ | ▼ | ? Close

From: Aarnio Ari (NVO/Helsinki)

To: Patent-Agency Banner-Witcoff (EXT-RES/Washington)

Cc:

Subject: 19377 Entitled A METHOD FOR PERFORMING HANDOVER FOR MULTICAST SESSIONS IN U

Sent: 10/1/01 6:02 PM **Importance:** Normal

Banner & Witcoff Ltd

1001 G Street, N.W.
Washington, DC 200001-4597
USA

— Reply by Tues w/sched/atty/etc.

01.10.2001
Re: Intended U.S. Patent Application in the name of Nokia Corporation
Entitled A METHOD FOR PERFORMING HANDOVER FOR MULTICAST SESSIONS IN U
Your Ref:
Our Ref: 19377
Rating: 2S
Inventor:Toni Paila, Everstinkuja 1 c 66, 02600 ESPOO
Jani Poikela, Kaarikuja 4 F 125, 00940 HELSINKI
Lin Xu, Vilppulanpolku 4 A 1, 33720 Tampere
Juha-Pekka Luoma, Sammonkatu 8 C 36, 33540 TAMPERE
Rod Walsh, Mäentakusenkatu 17 A 3 33710 TAMPERE

Brad,

I hope you are able to draft this new application.

BR Ari bc-ho.doc oki handover US 6259683..htm 19377 SEARCH A METHOD FOR PERFORMING HANDOVER FOR MULTICAST SESSIONS IN UNI-DIRECTIONAL ACCESS.htm

Close

From:	Patent-Agency Banner-Witcoff (EXT-RES/Washington)
To:	Aarnio Ari (NVO/Helsinki)
Cc:	
Subject:	NC19377; B&W 4770.00026 - First Draft
Sent:	10/30/01 10:06 PM

Importance: Normal

Ari,

Attached please find a first draft application (13 pages, inluding claims and abstract) and figures (5 additional pages, figures 1-7) for the above-referenced matter. Please have the inventors review the draft and provide any comments or changes. Specifically, please have them at least answer the questions embedded in the application in [ALL CAPS IN BRACKETS]. As this Application has a file-by date of November 19, 2001, PLEASE PROVIDE ANY COMMENTS TO US BY NOVEMBER 12, 2001 so that we will have time to prepare the revised draft and return it to you for approval. As always, please do not hesitate to contact us with any questions. We look forward to receiving your comments soon.

Regards,
Ross Dannenberg

Banner & Witcoff, Ltd
1001 G Street, NW
Washington, DC 20001-4597
Direct: (202) 508-9153
Direct Fax: (202) 585-5908
Main: (202) 508-9100
Main Fax: (202) 508-9299
rdannenberg@bannerwitcoff.com

IMPORTANT/CONFIDENTIAL: This message contains information from the law firm of Banner & Witcoff, LTD. which may be privileged, confidential, or exempt from disclosure under applicable law. If the reader of this message is not the intended recipient, or the employee or agent responsible for delivering the message to the intended recipient, you are hereby notified that any dissemination, distribution, retention, archiving, or copying of the communication is strictly prohibited. If you have received this communication in error, please notify us immediately by return e-mail, telephone, or facsimile.

422773_1.DOC 427030_1.PDF

Close

From: Patent-Agency Banner-Witcoff (EXT-RES/Washington)

To: Aarnio Ari (NVO/Helsinki)

Cc:

Subject: NC19377; BW 04770.00026 - Revised Draft

Sent: 11/13/01 10:00 PM **Importance:** Normal

Ari,

Attached please find a revised draft application and drawings for the above-referenced case, as well as formal declaration and assignment documents. Please have the inventors review the revised draft and, assuming all is in order, sign and return the executed declaration and assignment documents.

In reviewing the revised draft and accompanying papers, please note:
1) Also attached is a separate document showing the changes made in "redline" format, to more easily demonstrate the revisions from the previous draft.

2) Please confirm the inventors' citizenship information in the attached declaration document, as we only received citizenship information for the first named inventor. Please make any necessary corrections on the attached declaration document.

3) Please confirm the inventors' address information. Specifically, Rod Walsh has a different address than that provided with a previous application on which he is a named inventor. Please make any necessary corrections on the attached declaration and assignment documents.

Please let me know if you have any questions or if any other changes are required. As this application is due to be filed by November 19, please let us know as soon as possible if changes are required. Otherwise, we will file the application as soon as we receive the executed documents. Thank you for allowing us to be of assistance, and we look forward to hearing from you soon.

Regards,
Ross

Ross Dannenberg
Banner & Witcoff, Ltd
1001 G Street, NW
Washington, DC 20001-4597
Direct: (202) 508-9153
Direct Fax: (202) 585-5908
Main: (202) 508-9100
Main Fax: (202) 508-9299
rdannenberg@bannerwitcoff.com

IMPORTANT/CONFIDENTIAL: This message contains information from the law firm of Banner & Witcoff, LTD. which may be privileged, confidential, or exempt from disclosure under applicable law. If the reader of this message is not the intended recipient, or the employee or agent responsible for delivering the message to the intended recipient, you are hereby notified that any dissemination, distribution, retention, archiving, or copying of the communication is strictly prohibited. If you have received this communication in error, please notify us immediately by return e-mail, telephone, or facsimile.

Application - Redlined.DOC 427030 1.PDF 429489 1.DOC 429870 1.DOC 429871_1.DOC